

# **MELSEC A/Q**

Программируемые логические контроллеры

Руководство по программированию



**Руководство по программированию  
контроллеров MELSEC серии "A" и "Q"  
Артикул: 212580**

<b>Версия</b>			<b>Изменения/дополнения/исправления</b>
A	09/98	pdp	Руководство по программированию контроллеров MELSEC серий "A" и "Q" на основе Melsec Medoc plus
B	06/99	pdp	Раздел 5.1.2: изменена иллюстрация релейно-контактной схемы команды ORP изменена иллюстрация вариантов функционирования команды LDP Раздел 7.6.3: дополнено примечание, касающееся применения команды CALL Раздел 7.14: добавлены команды RSET_K_MD и RSET_K_P_MD
C	06/00	pdp	Раздел 7.11.13: дополнено примечание, касающееся применения команд ASC и ASCP
D	10/00	pdp	Отображение команд в GX Developer Дополнения для серии "Q" (Q02, Q02H, Q06H, Q12H и Q25H)
E	05/02	pdp-dk	Изменения названия: с "Melsec Medoc plus " на "GX IEC Developer", с "GPP/WIN" на "GX Developer" Дополнения для модулей центральных процессоров Q00J, Q00 и Q01 Новые команды для многопроцессорного режима: S.TO и FROM Дополнения к командам FREAD и FWRITE Дополнение специальных маркеров и регистров для центральных процессоров "System Q", начиная с версии "B" Раздельные перечни времени обработки для серии "A" и QnA/"System Q" Раздел 6.5.1: коррекция в источниках ошибок Раздел 6.5.2: коррекция в источниках ошибок Раздел 6.7.3: расширение описания команды COM для многопроцессорного режима Раздел 6.8.9: коррекция значений времени для n1 Раздел 7.1.1: коррекция операндов для MELSEC Q Раздел 7.1.3: коррекция операндов для MELSEC Q Раздел 7.1.5: коррекция операндов для MELSEC Q Раздел 7.1.7: коррекция операндов для MELSEC Q Раздел 7.5.12: коррекция источников ошибок Раздел 9: отображение команд в GX IEC Developer Раздел 9.5.1: коррекция значений времени при сравнении команд RBMOV и BMOV
F	09/04	pdp-dk	Новый раздел 10: команды для Q4AR Новый раздел 11: команды для специальных модулей Раздел 2.8: краткое описание команд для Q4AR Раздел 2.9: краткое описание команд для специальных модулей Новые модули центральных процессоров Q12PH и Q25PH



# Об этом руководстве

Содержащиеся в этом руководстве тексты, изображения, диаграммы и примеры предназначены исключительно для того, чтобы разъяснить программирование и применение программируемых контроллеров MELSEC серий "A", "Q" и "System Q".

Если у вас возникнут вопросы по программированию и эксплуатации, описанной в этом руководстве аппаратуры, обратитесь в региональное торговое представительство или к дилеру. Актуальную информацию и ответы на часто задаваемые вопросы вы можете найти в интернете ([www.mitsubishi-automation.ru](http://www.mitsubishi-automation.ru)).

MITSUBISHI ELECTRIC EUROPE BV сохраняет за собой право на внесение технических или иных изменений в это руководство, в любой момент и без уведомления.



# Содержание

<b>1</b>	<b>Введение</b>	
1.1	Прочие руководства .....	1-1
1.2	Типы центральных процессоров .....	1-2
1.3	Программное обеспечение .....	1-2
1.4	Нахождение команд .....	1-3
1.5	Параметры контроллеров .....	1-3
1.6	Сравнение GX IEC Developer и GX Developer .....	1-4
<b>2</b>	<b>Команды</b>	
2.1	Подразделение команд .....	2-1
2.2	Обзор команд .....	2-4
2.2.1	Пояснения к обзорной таблице .....	2-4
2.3	Базовый набор команд .....	2-6
2.3.1	Входные команды .....	2-6
2.3.2	Логические операции .....	2-7
2.3.3	Выходные команды .....	2-8
2.3.4	Команды сдвига .....	2-8
2.3.5	Организирующие команды .....	2-9
2.3.6	Команды конца программы .....	2-9
2.3.7	Прочие команды .....	2-9
2.4	Прикладные команды I .....	2-10
2.4.1	Операторы сравнения .....	2-10
2.4.2	Арифметические команды .....	2-15
2.4.3	Команды конвертирования .....	2-22
2.4.4	Команды передачи .....	2-25
2.4.5	Команды разветвления программы .....	2-27
2.4.6	Команды для вызова программ прерывания .....	2-27
2.4.7	Команда обновления данных .....	2-28
2.4.8	Прочие команды .....	2-29
2.5	Прикладные команды, часть II .....	2-31
2.5.1	Логические команды .....	2-31
2.5.2	Команды вращения .....	2-35
2.5.3	Команды сдвига .....	2-36
2.5.4	Команды обработки битов .....	2-37
2.5.5	Команды обработки данных .....	2-38

2.5.6	Команды структурирования программы .....	2-41
2.5.7	Команды для работы с перечнями данных.....	2-43
2.5.8	Команды обращения к буферной памяти .....	2-44
2.5.9	Команды индикации.....	2-45
2.5.10	Распознавание и устранение ошибок.....	2-46
2.5.11	Команды для обработки строковых величин .....	2-48
2.5.12	Команды особых функций .....	2-52
2.5.13	Команды контроля данных .....	2-55
2.5.14	Команды для переключения блоков регистров файлов .....	2-56
2.5.15	Команды для часов .....	2-57
2.5.16	Команды для периферийной аппаратуры .....	2-58
2.5.17	Программные команды.....	2-58
2.5.18	Прочие команды .....	2-59
2.6	Команды сетевой передачи данных .....	2-61
2.6.1	Команды обновления сетевых данных.....	2-61
2.6.2	Специализированные команды передачи данных (совместимые с серией QnA) .....	2-61
2.6.3	Команды передачи данных (совместимые с серией "A") .....	2-62
2.6.4	Информация маршрутизации .....	2-62
2.7	Команды для центральных процессоров MELSEC серии "System Q" .....	2-63
2.7.1	Считывание информации модуля .....	2-63
2.7.2	Отладка и поиск ошибок.....	2-63
2.7.3	Запись в файлы и считывание из файлов .....	2-64
2.7.4	Программные команды.....	2-64
2.7.5	Команды передачи .....	2-65
2.7.6	Команды для обмена данными в многопроцессорном режиме.....	2-65
2.8	Специальные команды для процессора Q4AR.....	2-66
2.8.1	Команды установки режима .....	2-66
2.8.2	Команды передачи .....	2-66
2.9	Команды для специальных модулей.....	2-67
2.9.1	Команды для модулей последовательных интерфейсов.....	2-67
2.9.2	Команды для модулей Profibus/DP.....	2-68
2.9.3	Команды для модулей Ethernet .....	2-69
2.9.4	Команда для MELSECNET/10 .....	2-69
2.9.5	Команды для CC-Link .....	2-70
<b>3      Конфигурация команд</b>		
3.1	Структура команды.....	3-1
3.1.1	Источник данных (s) .....	3-1
3.1.2	Место назначения (d).....	3-2
3.1.3	Количество (n).....	3-2



3.2	Обозначения, принятые в названиях команд .....	3-3
3.2.1	16/32-битные и импульсные команды .....	3-3
3.2.2	MELSEC и МЭК (IEC) .....	3-3
3.2.3	Прочие особенности написания .....	3-5
3.2.4	Написание команд в этом руководстве .....	3-5
3.3	Программирование специализированных команд .....	3-6
3.4	Программирование переменных .....	3-7
3.5	Типы данных .....	3-9
3.5.1	Обработка данных .....	3-11
3.5.2	Адресация массивов и регистров в GX IEC Developer .....	3-19
3.5.3	Применение строковых данных (STRING) .....	3-22
3.6	Индексация .....	3-24
3.6.1	Индексация .....	3-24
3.6.2	Особенности процессоров Q и QnA .....	3-26
3.6.3	Особенности процессоров AnA, AnAS и AnU .....	3-28
3.7	Косвенная адресация (только в GX Developer) .....	3-29
3.8	Ошибки обработки .....	3-31
3.8.1	Проверка области операндов .....	3-31
3.8.2	Проверка данных операндов .....	3-33
3.9	Условия выполнения команд .....	3-34
3.9.1	Входное условие .....	3-34
3.9.2	Вход EN и выход ENO .....	3-35
3.10	Количество шагов программы .....	3-37
3.10.1	В случае процессоров "System Q" и QnA .....	3-37
3.10.2	В случае процессоров AnA, AnAS и AnU .....	3-38

## **4 Структура разделов**

4.1	Обзор команд .....	4-2
4.2	Таблица процессоров .....	4-2
4.3	Операнды MELSEC A .....	4-3
4.4	Операнды MELSEC Q .....	4-4
4.4.1	Отображение в GX IEC Developer .....	4-4
4.4.2	Отображение в GX Developer .....	4-5
4.5	Переменные .....	4-5
4.6	Принцип действия .....	4-6
4.7	Примечания .....	4-6

4.8	Источники ошибок .....	4-6
4.9	Примеры .....	4-7
<b>5</b>	<b>Базовый набор команд</b>	
5.1	Входные команды .....	5-4
5.1.1	LD, LDI, AND, ANI, OR, ORI .....	5-4
5.1.2	LDP, LDF, ANDP, ANDF, ORP, ORF .....	5-8
5.2	Соединяющие команды .....	5-11
5.2.1	ANB, ORB .....	5-11
5.2.2	MPS, MRD, MPP .....	5-14
5.2.3	INV .....	5-17
5.2.4	MEP, MEF .....	5-19
5.2.5	EGP, EGF .....	5-21
5.3	Команды для выходных контактов .....	5-23
5.3.1	OUT .....	5-23
5.3.2	OUT T, OUTH T .....	5-25
5.3.3	OUT C .....	5-28
5.3.4	OUT F .....	5-31
5.3.5	SET .....	5-34
5.3.6	RST .....	5-36
5.3.7	SET F, RST F .....	5-39
5.3.8	PLS, PLF .....	5-42
5.3.9	FF .....	5-46
5.3.10	CHK .....	5-48
5.3.11	DELTA, DELTAP .....	5-50
5.4	Команды сдвига .....	5-52
5.4.1	SFT, SFTP .....	5-52
5.5	Организирующие команды .....	5-55
5.5.1	MC, MCR .....	5-55
5.6	Указание конца программы .....	5-61
5.6.1	FEND .....	5-61
5.6.2	END .....	5-64
5.7	Прочие команды .....	5-67
5.7.1	STOP .....	5-67
5.7.2	NOP .....	5-70

<b>6</b>	<b>Прикладные команды, часть I</b>	
6.1	Операторы сравнения.....	6-2
6.1.1	=, < >, >, <=, <, >= .....	6-5
6.1.2	D =, D < >, D >, D <=, D <, D >= .....	6-8
6.1.3	E =, E < >, E >, E <=, E <, E >= .....	6-11
6.1.4	\$ =, \$ < >, \$ >, \$ <=, \$ <, \$ >= .....	6-15
6.1.5	BKCMP, BKCMPP .....	6-20
6.2	Арифметические операции.....	6-25
6.2.1	+, + P, -, - P .....	6-28
6.2.2	D +, D + P, D -, D - P .....	6-32
6.2.3	x, xP, /, /P .....	6-36
6.2.4	D x, D x P, D/, D/P .....	6-40
6.2.5	B +, B + P, B -, B - P .....	6-43
6.2.6	DB +, DB + P, DB -, DB - P .....	6-48
6.2.7	B x, B x P, B/, B/P .....	6-53
6.2.8	DB x, DB x P, DB/, DB/P .....	6-56
6.2.9	E +, E + P, E -, E - P .....	6-60
6.2.10	E x, E x P, E/, E/P .....	6-65
6.2.11	BK +, BK + P, BK -, BK - P .....	6-68
6.2.12	\$ +, \$ + P .....	6-72
6.2.13	INC, INCP, DEC, DECP .....	6-75
6.2.14	DINC, DINCP, DDEC, DDECP .....	6-78
6.3	Команды конвертирования.....	6-81
6.3.1	BCD, BCDP, DBCD, DBCDP .....	6-83
6.3.2	BIN, BINP, DBIN, DBINP .....	6-86
6.3.3	FLT, FLTP, DFLT, DFLTP .....	6-90
6.3.4	INT, INTP, DINT, DINTP .....	6-94
6.3.5	DBL, DBLP .....	6-98
6.3.6	WORD, WORDP .....	6-100
6.3.7	GRY, GRYP, DGRY, DGRYP .....	6-102
6.3.8	GBIN, GBINP, DGBIN, DGBINP .....	6-105
6.3.9	NEG, NEGP, DNEG, DNEGP .....	6-108
6.3.10	ENEG, ENEGP .....	6-111
6.3.11	BKBCD, BKBCDP .....	6-113
6.3.12	BKBIN, BKBINP .....	6-116
6.4	Команды передачи .....	6-119
6.4.1	MOV, MOV P, DMOV, DMOV P .....	6-120
6.4.2	EMOV, EMOV P .....	6-123
6.4.3	\$MOV, \$MOV P .....	6-126
6.4.4	CML, CMLP, DCML, DCMLP .....	6-129
6.4.5	BMOV, BMOV P .....	6-134
6.4.6	FMOV, FMOV P .....	6-137
6.4.7	XCH, XCHP, DXCH, DXCHP .....	6-140

6.4.8	BXCH, BXCHP .....	6-143
6.4.9	SWAP, SWAPP .....	6-146
6.5	Команды разветвления программы .....	6-149
6.5.1	CJ, SCJ, JMP .....	6-150
6.5.2	GOEND .....	6-155
6.6	Команды для вызова программ прерывания .....	6-157
6.6.1	DI, EI, IMASK .....	6-158
6.6.2	IRET .....	6-165
6.7	Команды обновления данных .....	6-167
6.7.1	RFS, RFSP .....	6-168
6.7.2	SEG .....	6-170
6.7.3	COM .....	6-174
6.7.4	EI, DI .....	6-177
6.8	Прочие команды .....	6-180
6.8.1	UDCNT1 .....	6-181
6.8.2	UDCNT2 .....	6-184
6.8.3	TTMR .....	6-187
6.8.4	STMR, STMRH .....	6-189
6.8.5	ROTC .....	6-193
6.8.6	RAMP .....	6-198
6.8.7	SPD .....	6-200
6.8.8	PLSY .....	6-202
6.8.9	PWM .....	6-204
6.8.10	MTR .....	6-206

## **7 Прикладные команды, часть II**

7.1	Логические команды .....	7-2
7.1.1	WAND, WANDP, DAND, DANDP .....	7-4
7.1.2	BKAND, BKANDP .....	7-11
7.1.3	WOR, WORP, DOR, DORP .....	7-14
7.1.4	BKOR, BKORP .....	7-20
7.1.5	WXOR, WXORP, DXOR, DXORP .....	7-23
7.1.6	BKXOR, BKXORP .....	7-29
7.1.7	WXNR, WXNRP, DXNR, DXNRP .....	7-32
7.1.8	BKXNR, BKXNRP .....	7-39
7.2	Команды вращения битов .....	7-42
7.2.1	ROR, RORP, RCR, RCRP .....	7-43
7.2.2	ROL, ROLP, RCL, RCLP .....	7-46
7.2.3	DROR, DRORP, DRORP, DRORP .....	7-49
7.2.4	DROL, DROLP, DRCL, DRCLP .....	7-52

---

7.3	Команды сдвига .....	7-55
7.3.1	SFR, SFRP, SFL, SFLP .....	7-56
7.3.2	BSFR, BSFRP, BSFL, BSFLP .....	7-59
7.3.3	DSFR, DSFRP, DSFL, DSFLP .....	7-62
7.4	Команды обработки битов .....	7-65
7.4.1	BSET, BSETP, BRST, BRSTP .....	7-66
7.4.2	TEST, TESTP, DTEST, DTESTP.....	7-69
7.4.3	BKRST, BKRSTP .....	7-73
7.5	Команды обработки данных.....	7-77
7.5.1	SER, SERP, DSER, DSERP .....	7-79
7.5.2	SUM, SUMP, DSUM, DSUMP .....	7-85
7.5.3	DECO, DECOP .....	7-88
7.5.4	ENCO, ENCO P .....	7-90
7.5.5	SEG, SEGP .....	7-92
7.5.6	DIS, DISP .....	7-96
7.5.7	UNI, UNIP .....	7-99
7.5.8	NDIS, NDISP, NUNI, NUNIP .....	7-102
7.5.9	WTOB, WTOBP, BTOW, BTOWP .....	7-107
7.5.10	MAX, MAXP, DMAX, DMAXP.....	7-113
7.5.11	MIN, MINP, DMIN, DMINP .....	7-116
7.5.12	SORT, SORTP, DSORT, DSORTP .....	7-119
7.5.13	WSUM, WSUMP.....	7-123
7.5.14	DWSUM, DWSUMP .....	7-125
7.6	Команды для структурирования программы .....	7-127
7.6.1	FOR, NEXT .....	7-128
7.6.2	BREAK, BREAKP .....	7-131
7.6.3	CALL, CALLP .....	7-134
7.6.4	RET.....	7-137
7.6.5	FCALL, FCALLP .....	7-139
7.6.6	ECALL, ECALLP .....	7-143
7.6.7	EFCALL, EFCALLP .....	7-146
7.6.8	CHG.....	7-150
7.6.9	SUB, SUBP .....	7-159
7.6.10	IX, IXEND .....	7-162
7.6.11	IXDEV, IXSET .....	7-167
7.7	Команды для работы с перечнями данных .....	7-170
7.7.1	FIFW, FIFWP .....	7-171
7.7.2	FIFR, FIFRP.....	7-175
7.7.3	FPOP, FPOPP .....	7-179
7.7.4	FDEL, FDELP, FINS, FINS P.....	7-183
7.8	Команды доступа к буферной памяти .....	7-189
7.8.1	FROM, FROMP, DFRO, DFROP .....	7-190
7.8.2	TO, TOP, DTO, DTO P .....	7-194

7.9	Команды индикации.....	7-198
7.9.1	PR.....	7-200
7.9.2	PRC.....	7-206
7.9.3	LED.....	7-211
7.9.4	LEDC.....	7-214
7.9.5	LEDA, LEDB.....	7-217
7.9.6	LEDR.....	7-219
7.10	Выявление и устранение ошибок.....	7-223
7.10.1	CHKST, CHK (серии "Q"/"System Q").....	7-224
7.10.2	CHK (серия "A").....	7-232
7.10.3	CHKCIR, CHKEND.....	7-240
7.10.4	SLT, SLTR.....	7-245
7.10.5	STRA, STRAR.....	7-248
7.10.6	PTRA, PTRAR, PTRAEHE, PTRAEHEP.....	7-251
7.11	Команды для обработки строковых величин.....	7-253
7.11.1	BINDA, BINDAP, DBINDA, DBINDAP.....	7-256
7.11.2	BINHA, BINHAP, DBINHA, DBINHAP.....	7-261
7.11.3	BCDDA, BCDDAP, DBCDDA, DBCDDAP.....	7-266
7.11.4	DABIN, DABINP, DDABIN, DDABINP.....	7-271
7.11.5	HABIN, HABINP, DHABIN, DHABINP.....	7-276
7.11.6	DABCD, DABCDP, DDABCD, DDABCDP.....	7-280
7.11.7	COMRD, COMRDP.....	7-285
7.11.8	LEN, LENP.....	7-289
7.11.9	STR, STRP, DSTR, DSTRP.....	7-292
7.11.10	VAL, VALP, DVAL, DVALP.....	7-299
7.11.11	ESTR, ESTRP.....	7-305
7.11.12	EVAL, EVALP.....	7-314
7.11.13	ASC, ASCP (серии "Q" и "System Q").....	7-320
7.11.14	ASC (серия "A").....	7-323
7.11.15	HEX, HEXP.....	7-325
7.11.16	RIGHT, RIGHTP, LEFT, LEFTP.....	7-329
7.11.17	MIDR, MIDRP, MIDW, MIDWP.....	7-333
7.11.18	INSTR, INSTRP.....	7-339
7.11.19	EMOD, EMODP.....	7-343
7.11.20	EREXP, EREXPP.....	7-347
7.12	Особые функции.....	7-350
7.12.1	SIN, SINP.....	7-352
7.12.2	COS, COSP.....	7-355
7.12.3	TAN, TANP.....	7-358
7.12.4	ASIN, ASINP.....	7-361
7.12.5	ACOS, ACOSP.....	7-364
7.12.6	ATAN, ATANP.....	7-367
7.12.7	RAD, RADP.....	7-370
7.12.8	DEG, DEGP.....	7-373

7.12.9	SQR, SQRP .....	7-376
7.12.10	EXP, EXPP .....	7-379
7.12.11	LOG, LOGP .....	7-382
7.12.12	RND, RNDP, SRND, SRNDP .....	7-385
7.12.13	BSQR, BSQRP, BDSQR, BDSQRP .....	7-387
7.12.14	BSIN, BSINP .....	7-391
7.12.15	BCOS, BCOSP .....	7-394
7.12.16	BTAN, BTANP .....	7-397
7.12.17	BASIN, BASINP .....	7-400
7.12.18	BACOS, BACOSP .....	7-403
7.12.19	BATAN, BATANP .....	7-406
7.13	Команды контроля данных .....	7-409
7.13.1	LIMIT, LIMITP, DLIMIT, DLIMITP .....	7-410
7.13.2	BAND, BANDP, DBAND, DBANDP .....	7-414
7.13.3	ZONE, ZONEP, DZONE, DZONEP .....	7-418
7.14	Команды для переключения блоков регистров файлов .....	7-422
7.14.1	RSET, RSETP .....	7-423
7.14.2	QDRSET, QDRSETP .....	7-426
7.14.3	QCDSSET, QCDSSETP .....	7-429
7.15	Команды для часов .....	7-432
7.15.1	DATERD, DATERDP .....	7-433
7.15.2	DATEWR, DATEWRP .....	7-438
7.15.3	DATE+, DATE+P .....	7-443
7.15.4	DATE-, DATE-P .....	7-448
7.15.5	SECOND, SECONDP, HOUR, HOURP .....	7-453
7.16	Команды для периферийных устройств .....	7-459
7.16.1	MSG .....	7-460
7.16.2	PKEY .....	7-463
7.17	Команды для управления программой .....	7-466
7.17.1	PSTOP, PSTOPP .....	7-467
7.17.2	POFF, POFFP .....	7-469
7.17.3	PSCAN, PSCANP .....	7-471
7.17.4	PLOW, PLOWP .....	7-473
7.18	Прочие команды .....	7-475
7.18.1	WDT, WDTP .....	7-476
7.18.2	STC, CLC .....	7-478
7.18.3	DUTY .....	7-480
7.18.4	ZRRDB, ZRRDBP .....	7-483
7.18.5	ZRWRB, ZRWRBP .....	7-487
7.18.6	ADRSET, ADRSETP .....	7-491
7.18.7	KEY .....	7-492
7.18.8	ZPUSH, ZPUSHP, ZPOP, ZPOPP .....	7-498
7.18.9	EROMWR, EROMWRP .....	7-501

<b>8</b>	<b>Команды коммуникации</b>	
8.1	Основные сведения .....	8-1
8.2	Типы команд .....	8-1
8.3	Области записи и считывания данных .....	8-3
8.3.1	MELSECNET/10 .....	8-3
8.3.2	MELSECNET .....	8-4
8.4	Специализированные команды коммуникации .....	8-4
8.4.1	Одновременное выполнение .....	8-4
8.4.2	Конец передачи данных .....	8-4
8.5	Команды обновления данных .....	8-6
8.5.1	ZCOM .....	8-7
8.6	Специализированные команды коммуникации для серии QnA .....	8-11
8.6.1	READ .....	8-12
8.6.2	SREAD .....	8-19
8.6.3	WRITE .....	8-26
8.6.4	SWRITE .....	8-33
8.6.5	SEND .....	8-40
8.6.6	RECV .....	8-48
8.6.7	REQ .....	8-53
8.6.8	ZNFR .....	8-65
8.6.9	ZNTO .....	8-71
8.7	Команды коммуникации, совместимые с серией "A" .....	8-77
8.7.1	ZNRD .....	8-78
8.7.2	ZNWR .....	8-82
8.7.3	LRDP .....	8-86
8.7.4	LWTP .....	8-90
8.7.5	RFRP .....	8-94
8.7.6	RTOP .....	8-100
8.8	Считывание и запись информации маршрутизации .....	8-106
8.8.1	RTREAD .....	8-107
8.8.2	RTWRITE .....	8-109
<b>9</b>	<b>Команды для процессоров "System Q"</b>	
9.1	Считывание информации модуля .....	9-2
9.1.1	UNIRD, UNIRDP .....	9-2
9.2	Поиск и устранение ошибок .....	9-7
9.2.1	TRACE, TRACER .....	9-7



9.3	Передача данных в файлы и из файлов .....	9-9
9.3.1	FWRITE .....	9-9
9.3.2	FREAD .....	9-20
9.4	Команды обращения с программой .....	9-33
9.4.1	PLOADP .....	9-33
9.4.2	PUNLOADP .....	9-36
9.4.3	PSWAPP .....	9-38
9.5	Команды передачи .....	9-41
9.5.1	RBMOV, RBMOV P .....	9-41
9.6	Команды для многопроцессорного режима .....	9-46
9.6.1	S.TO, SP.TO .....	9-46
9.6.2	FROM, FROM P .....	9-49

## **10 Команды для Q4ARCPU**

10.1	Команды выбора режима .....	10-2
10.1.1	STMODE .....	10-2
10.1.2	CGMODE .....	10-4
10.2	Команды передачи данных .....	10-6
10.2.1	TRUCK .....	10-6
10.2.2	SPREF .....	10-11

## **11 Команды для специальных модулей**

11.1	Команды для модулей последовательного интерфейса .....	11-2
11.1.1	BUFRCVS .....	11-3
11.1.2	GETE, GETEP .....	11-6
11.1.3	PUTE, PUTEP .....	11-11
11.1.4	PRR, PRRP .....	11-18
11.2	Команды для модулей Profibus/DP .....	11-26
11.2.1	BBLKRD, BBLKRDP .....	11-27
11.2.2	BBLKWR, BBLKWRP .....	11-30
11.3	Команды для модулей Ethernet .....	11-33
11.3.1	BUFRCV .....	11-34
11.3.2	BUFRCVS .....	11-39
11.3.3	BUFSND .....	11-42
11.3.4	OPEN .....	11-47
11.3.5	CLOSE .....	11-56
11.3.6	ERRCLR .....	11-61
11.3.7	ERRRD .....	11-67
11.3.8	UINI .....	11-72

11.4	Команды для MELSECNET/10.....	11-78
11.4.1	PAIRSET .....	11-79
11.5	Команды для CC-Link .....	11-82
11.5.1	RLPA (серия "A").....	11-83
11.5.2	RLPASET (System Q) .....	11-90
11.5.3	RRPA (серия "A") .....	11-102
11.5.4	RIRD (серия "A") .....	11-109
11.5.5	RIRD (серии QnA и "System Q") .....	11-115
11.5.6	RIWT (серия "A").....	11-123
11.5.7	RIWT (серии QnA и "System Q").....	11-129
11.5.8	RIRCV (серия "A").....	11-137
11.5.9	RIRCV (серии QnA и "System Q") .....	11-143
11.5.10	RISEND (серия "A") .....	11-149
11.5.11	RISEND (серии QnA и "System Q") .....	11-155
11.5.12	RITO (серия "A") .....	11-161
11.5.13	RITO (серии QnA и "System Q") .....	11-165
11.5.14	RIFR (серия "A") .....	11-169
11.5.15	RIFR (серии QnA и "System Q").....	11-173

## **12 Программа микрокомпьютера (AnN(S))**

12.1	Объем и области памяти .....	12-1
12.2	Применение самостоятельно созданных программ микрокомпьютера .....	12-2
12.2.1	Распределение памяти .....	12-3
12.2.2	Структура адресов в области для сохранения данных .....	12-3
12.2.3	Разбивка области памяти .....	12-4

## **13 Коды ошибок**

13.1	Перечень кодов ошибок (процессоры Q00J, Q00 и Q01).....	13-2
13.2	Перечень кодов ошибок (серии QnA и "System Q") .....	13-12
13.3	Перечень кодов ошибок серии "A" (кроме AnA и AnAS) .....	13-38
13.4	Перечень кодов ошибок серий AnA и AnAS .....	13-42

## **14 Приложение А**

A.1	Определение времени обработки.....	A-1
A.2	Значения времени обработки .....	A-2
A.2.1	Перечень значений времени обработки (серии QnA и System Q) .....	A-3
A.2.2	Время обработки в контроллерах MELSEC серии "A" .....	A-24

---

A.3	Сравнение центральных процессоров .....	A-33
A.3.1	Применимые операнды .....	A-33
A.3.2	Режимы обработки ввода-вывода .....	A-35
A.3.3	Типы данных .....	A-35
A.3.4	Сравнение таймеров .....	A-36
A.3.5	Сравнение счетчиков .....	A-39
A.3.6	Сравнение команд индикации .....	A-40
A.3.7	Команды серий "Q" и "System Q", эквивалентные командам MELSEC серии "A" .....	A-41
A.3.8	Сравнение процессоров QnA/Q2AS с процессорами MELSEC System Q .....	A-42
A.4	Обзор специальных маркеров .....	A-44
A.4.1	Перечень маркеров диагностики (серии "MELSEC Q" и "MELSEC System Q") .....	A-44
A.4.2	Перечень специальных маркеров (серии "A") .....	A-66
A.4.3	Обзор специальных маркеров связи (только серия "A") .....	A-72
A.5	Обзор специальных регистров .....	A-75
A.5.1	Обзор регистров диагностики (MELSEC серии "Q" и "System Q") .....	A-75
A.5.2	Специальные регистры (только MELSEC серии "A") .....	A-112
A.5.3	Обзор специальных регистров связи (только MELSEC серии "A") .....	A-122



# 1 Введение

В этом руководстве описаны программирование и обработка базовых и прикладных команд, имеющихся в центральных процессорах MELSEC серий "A", QnA и "System Q".

## 1.1 Прочие руководства

Руководство по программированию контроллеров MELSEC серий QnA и "System Q" (команды регулирования)

– описание команд для реализации ПИД-регуляторов

Programming Manual (AD57/58)

– описание специальных команд для специальных модулей AD57/58

Programming Manual MELSEC QnA Series and MELSEC System Q (SFC)

– описание команд для программирования на языке SFC

Руководство по GX Developer

– описание онлайн-функций среды GX Developer, включая программирование и поиск ошибок

GX IEC Developer Beginners Manual/Пособие для начинающего

– основы программирования с помощью GX IEC Developer

GX IEC Developer Reference Manual/Справочник по программированию

– подробное описание программирования с помощью GX IEC Developer

– описание стандартных команд IEC (стандартная библиотека МЭК)

**ПРИМЕЧАНИЕ** *Все руководства перечислены в нашем текущем прайс-листе на контроллеры и выложены для скачивания на сайте MITSUBISHI ELECTRIC ([www.mitsubishi-automation.ru](http://www.mitsubishi-automation.ru)) в виде файлов PDF.*

## 1.2 Типы центральных процессоров

Описываемые в этом руководстве функции можно с помощью среды программирования GX Developer или GX IEC Developer (в их последних версиях) передать на центральные процессоры всех типов, при условии, что процессоры поддерживают используемые команды.

Описываемые в руководстве команды действительны для следующих программируемых контроллеров MELSEC и типов центральных процессоров:

Тип контроллера		Типы центральных процессоров
Серия "A"	AnA/AnU	A2A, A2A-S1, A2U, A2U-S1, A3A, A3U
	AnAS/AnUS	A2AS, A2AS-S1, A2AS-S30, A2AS-S60, A2US, A2US-S1
	AnN	A1, A2, A2C A3M, A3N
	AnS	A1S, A1S-S1, A2S, A2S-S1
Серия "Q"	QnA	Q2A/Q2AS, Q2A-S1/Q2AS-S Q3A Q4A, Q4AR
System Q	Q (однопроцессорные модули центрального процессора)	Q00J
	Q (многопроцессорные модули центрального процессора)	Q00, Q01 (ограниченно пригоден для многопроцессорного режима) Q02, Q02H, Q06H, Q12H, Q12PH, Q25H, Q12PH Модули центр. процессора ПК: PPC-CPU686(MS)-64 PPC-CPU686(MS)-128

Если в тексте имеются общие упоминания "MELSEC A", "MELSEC Q" или просто "A" и "Q" (например, в таблицах), то это означает, что имеются в виду все типы центральных процессоров серии "A" или "Q" и "System Q". На исключения в тексте указывается особо.

## 1.3 Программное обеспечение

Все описанные в руководстве команды, – кроме небольшого числа исключений, – можно использовать в имеющихся пакетах программного обеспечения:

- GX Developer
- GX IEC Developer

Примеры этого руководства созданы в среде GX IEC Developer. Примеры на языке списка инструкций (IL) для контроллеров MELSEC всегда соответствуют среде GX Developer.

Все описываемые в этом руководстве команды входят в библиотеку среды программирования GX IEC Developer.

В диалоговом окне GX IEC Developer для выбора команд всегда отображаются только те команды, которые действительно способен обрабатывать выбранный центральный процессор.

## 1.4 Нахождение команд

### Для опытных пользователей

Если вы уже знакомы с программированием команд для контроллеров MELSEC серий "A", "Q" и "System Q", ищите требуемую информацию в разделах с 5 по 9, посвященных описанию команд. В верхнем колонтитуле страниц указано название команды в том виде, в каком она используется в среде GX Developer и редакторе MELSEC среды GX IEC Developer.

### Для начинающих пользователей

Если вы еще не очень хорошо ориентируетесь в работе с командами, действуйте следующим образом:

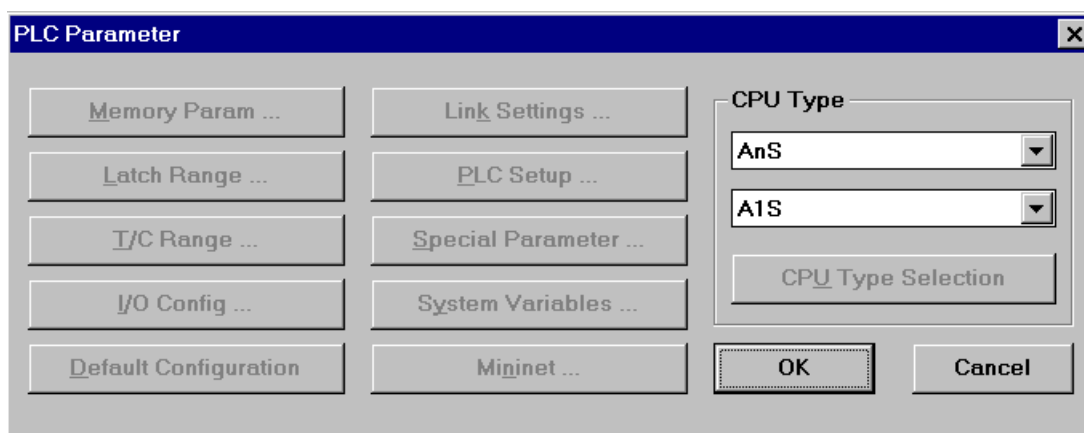
- Прочтите в разделе 3 о различиях в написании команд между редактором MELSEC и редактором стандартных команд МЭК (IEC).
- Прочтите пояснения в разделе 4, чтобы понять единую структуру описания каждой команды.
- Используйте
  - табличный обзор групп команд с их краткими описаниями в разделе 2,
  - указатель ключевых слов, в котором можно найти все команды.

**ПРИМЕЧАНИЕ** Все рассматриваемые в этом руководстве команды со столь же подробными описаниями можно найти и в оперативном справочнике среды программирования GX IEC Developer.

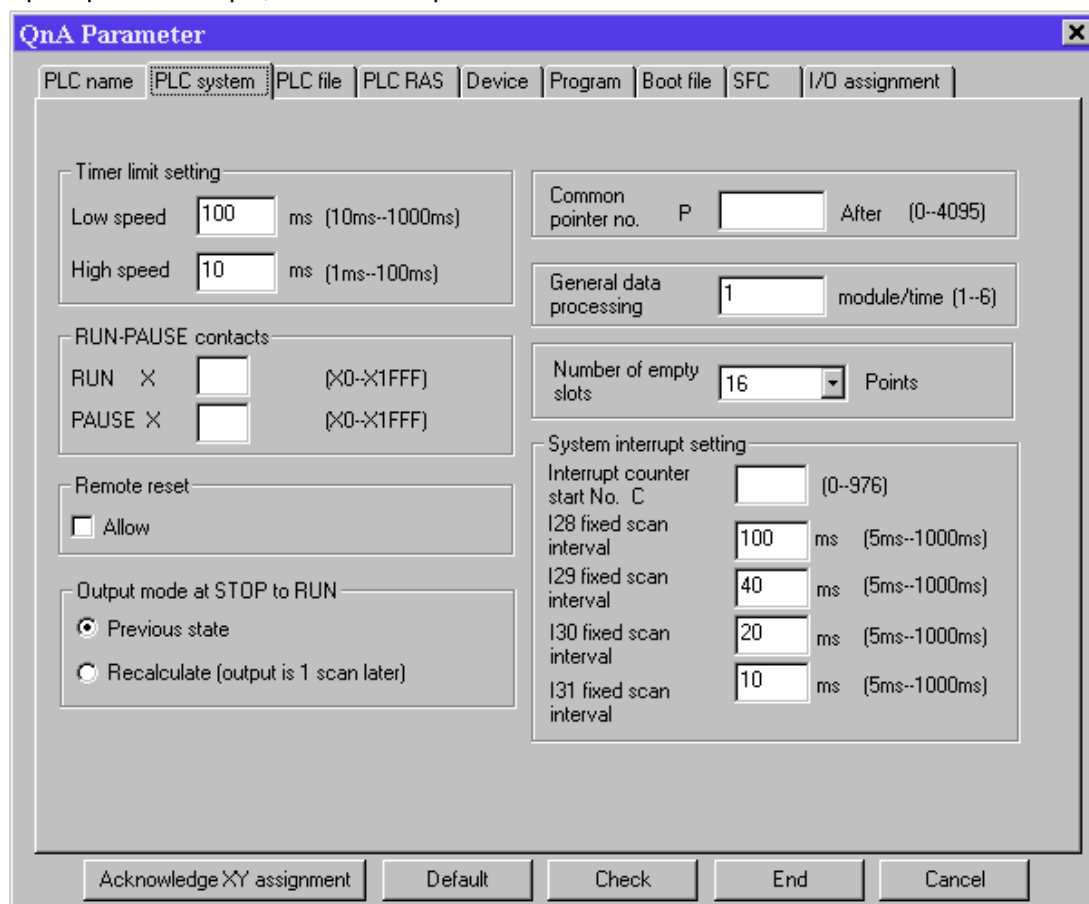
## 1.5 Параметры контроллеров

Параметры служат для задания определенных функций, областей операндов и т. п. Для программирования функций, описываемых в этом руководстве, параметры можно оставить без изменений или приспособить их в соответствии с потребностями пользователя. Более подробные указания по настройке параметров контроллера имеются в соответствующих описаниях аппаратной части модулей центральных процессоров, а также в руководствах по их программированию.

Пример: GX IEC Developer



Пример: GX Developer, GX IEC Developer 6.0



## 1.6 Сравнение GX IEC Developer и GX Developer

Следующая таблица содержит обзор наиболее важных свойств пакетов программного обеспечения GX IEC Developer и GX Developer.

GX IEC Developer	GX Developer
Структурированное использование	Простое использование
Программирование по стандарту IEC 1131	—
Имеющиеся редакторы: список инструкций (IL), релейно-контактная схема (LD), структурированный текст (ST), SFC, FUB	Имеющиеся редакторы: список инструкций (IL), релейно-контактная схема (LD), SFC
Функции и функциональные блоки	Функциональные блоки (начиная с V. 7)
Изменение программы в режиме онлайн	Изменение программы в режиме онлайн Возможна замена программ в режиме онлайн
Функции диагностики контроллера	Функции диагностики контроллера
Функции диагностики сетей	Функции диагностики сетей



## 2 Команды

### 2.1 Подразделение команд

Команды можно подразделить на несколько главных групп, перечисленных в следующей таблице:

Группы команд		Описание	Ссылка
Базовый набор команд	Входные команды	начало операции, последовательное и параллельное соединение контактов	разд. 5.1
	Логические операции	последовательное и параллельное соединение блоков, сохранение и обработка результата логической операции, инвертирование результатов операций, преобразование результатов операции в импульсные сигналы, установка маркеров фронта	разд. 5.2
	Выходные команды	битовые операнды, контакты счетчиков и таймеров, вывод, установка и сброс маркеров ошибки, установка и сброс операндов, вывод по фронту сигнала, инвертирование коммутационного состояния операнда, выработка переключающих импульсов	разд. 5.3
	Команды сдвига	перемещение битовых операндов	разд. 5.4
	Организирующие команды	активация и деактивация частей программы	разд. 5.5
	Команды конца программы	конец области программы, конец основных программ и подпрограмм	разд. 5.6
	Прочие команды	прерывание обработки, пустой шаг в программе	разд. 5.7
Прикладные команды, часть I	Операторы сравнения	сравнение данных, например, =, >, ≥ и т. п.	разд. 6.1
	Арифметические операции	сложение, вычитание, умножение, деление, двоичных и двоично-десятичных данных, чисел с плавающей запятой и двоичных блоков данных, логическое соединение строковых величин, приращение, декремент	разд. 6.2
	Команды конвертирования	конвертирование данных, например, BCD → BIN и BIN → BCD	разд. 6.3
	Команды передачи	передача, обмен данными и инверсия данных	разд. 6.4
	Команды разветвления программы	переход, вызов подпрограммы	разд. 6.5
	Вызов программ прерывания	вызов программ прерывания	разд. 6.6
	Команды обновления данных	обновление канала связи и интерфейса ввода-вывода	разд. 6.7
	Прочие команды	однофазный/двухфазный возрастающий/убывающий счетчик, программируемые таймеры, таймер особой функции, команда позиционирования, линейно изменяющийся сигнал, счетчик импульсов, вывод импульсов, широтно-импульсная модуляция, матрица ввода	разд. 6.8

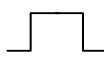
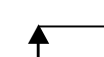
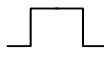

Группы команд		Описание	Ссылка
Прикладные команды, часть II	Логические операции	логика "И", "ИЛИ", "исключающее ИЛИ", "НЕ ИЛИ"	разд. 7.1
	Команды вращения	вращение 16-битных и 32-битных данных вправо/влево	разд. 7.2
	Команды сдвига	Побитовый или поблочный сдвиг в пределах слова данных	разд. 7.3
	Команды обработки битов	установка и сброс битов, опрос битов	разд. 7.4
	Команды обработки данных	поиск данных в определенных областях, кодировка и декодировка данных, разделение и соединение данных	разд. 7.5
	Команды структурирования программы	команда повторения, вызов подпрограммы, вызов подпрограммы в другом программном файле, переключение между областями Main и Sub программы, вызов программы микрокомпьютера, индексированная адресация области программы, сохранение индексированных адресов операндов в индексном перечне	разд. 7.6
	Команды для работы с перечнями данных	запись данных в перечень для дальнейшей обработки и их считывание из перечня, стирание определенных блоков данных в перечне и вставление новых блоков данных	разд. 7.7
	Команды доступа к буферной памяти	обращение к буферной памяти специального модуля или удаленного модуля	разд. 7.8
	Команды индикации	вывод знаков ASCII на выходы модуля или светодиодный дисплей центрального процессора	разд. 7.9
	Команды диагностики и отладки	контроль на ошибки, фиксация состояния, семплирование (выборочная трассировка), контроль программы (трассировка программы)	разд. 7.10
	Команды для обработки строковых величин	обработка строковых величин (ASCII-код)	разд. 7.11
	Команды для особых функций	команды для угловых функций, вычисления корней и степеней двоично-десятичных данных и чисел с плавающей запятой	разд. 7.12
	Команды контроля данных	проверка входных данных путем сравнения с заданными диапазонами значений и сохранение проверенных данных	разд. 7.13
	Команды переключения блоков регистров данных и файлов	переключение между блоками регистров файлов и файлами	разд. 7.14
	Команды для часов	запись и считывание данных времени	разд. 7.15
	Команды для периферийных устройств	вывод сообщений и ввод с клавиатуры на периферийных устройствах	разд. 7.16
	Программные команды	команды для изменения режимов выполнения программы	разд. 7.17
Прочие команды	сброс сторожевого таймера, установка и сброс флага переноса, выработка импульсов, непосредственное считывание и запись байтов, ввод с клавиатуры, сохранение и восстановление содержимого индексных регистров, запись данных в регистр EEPROM	разд. 7.18	

Группы команд		Описание	Ссылка
Команды коммуникации	Команды обновления сетевых данных	команды обновления данных в модулях сетевой коммуникации	разд. 8.5
	Специализированные команды коммуникации	считывание данных из целевых станций и их запись в целевые станции в целевых сетях, передача данных на модули сетевой коммуникации в целевых станциях в целевых сетях, считывание данных, переданных с помощью команды SEND, запрос данных от других станций (операции записи и считывания данных часов, операция RUN/STOP), считывание данных из специальных модулей и их запись в специальные модули в удаленных станциях ввода-вывода	разд. 8.6
	Команды коммуникации, совместимые с серией "A"	считывание данных из целевых станций и их запись в целевые станции в целевых сетях, запись данных в локальные станции и их считывание из локальных станций (только в главных станциях), считывание данных из специальных модулей и их запись в специальные модули в удаленных станциях ввода-вывода.	разд. 8.7
	Считывание и запись информации маршрутизации	считывание и запись параметров маршрутизации (номер сети, номер релейной станции, номер станции маршрутизации)	разд. 8.8
Команды для центрального процессора "System Q"	Считывание информации модуля	непосредственное считывание из памяти модуля	разд. 9.1
	Выявление и устранение ошибок	установка и сброс контроля (трассировки)	разд. 9.2
	Перенос данных в файлы и из файлов	запись данных в файлы, считывание данных из файлов	разд. 9.3
	Операции с программой	загрузка программы из памяти, стирание программы в памяти	разд. 9.4
	Команды передачи	передача данных	разд. 9.5
	Команды для обмена данными в многопроцессорном режиме	ввод данных в общую область памяти, считывание данных из общей области памяти другого центрального процессора	разд. 9.6
Команды для центрального процессора Q4AR	Выбор режима	выбор метода запуска центрального процессора и его поведения при переключении на резервный центральный процессор избыточной системы	разд. 10.1
	Команды передачи	передача данных из активного центрального процессора в резервный центральный процессор и обмен данными с буферной памятью одного или нескольких специальных модулей	разд. 10.2
Команды для специальных модулей	команды для модулей последовательного интерфейса	передача принятых данных в программе прерывания в центральный процессор, считывание, установление или стирание определенных пользователем кадров данных, передача данных с помощью определенных пользователем кадров данных	разд. 11.1
	Команды для модулей Profibus/DP	обмен данными с буферной памятью модуля Profibus	разд. 11.2
	Команды для модулей Ethernet	считывание принятых данных из фиксированного буфера, ввод данных в фиксированные буферы, установление и завершение сеанса связи, стирание сообщений об ошибках из буфера и выключение светодиода "ERR.", считывание кода ошибки из модуля Ethernet, повторная инициализация модуля Ethernet	разд. 11.3
	Команда для MELSECNET/10	установление станций, соединенных при дуплексном режиме	разд. 11.4
	Команды для CC-Link	настройка параметров сети, установление параметров автоматического обновления, считывание данных из буферной памяти модуля CC-Link или центрального процессора контроллера этой же станции или запись данных, считывание данных (с кэшированием) из буферной памяти интеллектуальной станции или их запись в буферную память, считывание данных из автоматически обновляемой области памяти или их запись в эту область	разд. 11.5

## 2.2 Обзор команд

### 2.2.1 Пояснения к обзорной таблице

В следующих разделах (с 2.3 по 2.6) дан обзор всех описываемых в этом руководстве команд. Обзорная таблица имеет следующие столбцы:

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Сложение и вычитание 16-битных двоичных данных	+	s, d	$(d) + (s) \rightarrow (d)$		3	5	6.2.1
	+P						
	+	s1, s2, d1	$(s1) + (s2) \rightarrow (d1)$		4	7	6.2.1
	+P						

↑ (1)      ↑ (2)      ↑ (3)      ↑ (4)      ↑ (5)      ↑ (6)      ↑ (7)      ↑ (8)

Разъяснение отдельных столбцов:

(1) Группа команд

(2) Указание названия команды для программирования

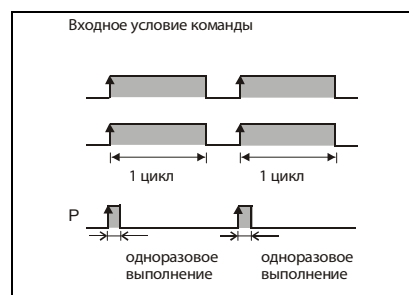
Название команды указывается в написании для контроллеров MELSEC (разъяснение этого написания см. в разделе 3.2).

Всегда указываются 16-битные команды. В случае 32-битных команд в начале названия команды добавляется буква "D".

- 16-битная команда: +
- 32-битная команда: D+

К импульсным командам, т. е. командам, выполняемым только при положительном или отрицательном фронте сигнала, добавляется буква "P".

- обычная команда: +
- импульсная команда: +P



Команды, обрабатывающие строковые величины, обозначаются символом "\$" в начале команды:  
 – обычная команда: +  
 – строковая команда: \$+

### (3) Указание применяемых переменных

Здесь указываются переменные. Источник данных обозначается буквой "s" (от англ. слова "source" – "источник"), а место назначения – буквой "d" (от англ. слова "destination" – "место назначения").

Пример: s = если имеется только один источник данных

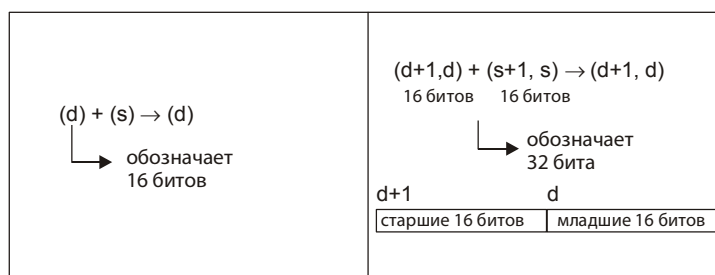
s1, s2 = если имеются несколько источников данных

s + 0, s + 1, (s1) + 0, (s1) + 1 = в случае 32-битных команд




Например, s 1 = регистр данных D0, (s1) + 1 = регистр данных D1

s + 0, s + 1, s + 2, s + 3 = 4 следующих друг за другом операнда, например, часть массива

### (4) Значение и обработка всей управляющей команды



### (5) Условие выполнения отображается в соответствии со следующей таблицей

Символ	Условие выполнения
без символа	Команда выполняется постоянно и независимо от состояния предшествующего условия. Если входное условие не установлено, команда не выполняется.
	Команда выполняется до тех пор, пока выполняется входное условие. Если входное условие перестало выполняться, команда более не выполняется и не обрабатывается.
	Эта команда представляет собой импульсную команду. Она выполняется только один раз, при положительном фронте входного сигнала (если состояние входного условия изменилось с "выключено" на "включено"). После этого команда не выполняется и не обрабатывается даже при наличии входного сигнала.
	Эта команда также представляет собой импульсную команду. Она выполняется только один раз, при отрицательном фронте входного сигнала (если состояние входного условия изменилось с "включено" на "выключено"). После этого команда не выполняется и не обрабатывается даже при наличии входного сигнала.

### (6 + 7) Указание количества шагов программы

В этом столбце указывается число шагов, необходимых для полного выполнения команды. Здесь имеется различие между контроллерами MELSEC серии "A" и серий "Q"/"System Q". Более подробные пояснения имеются в разделе 3.9.

### (8) Ссылка на раздел

Показывает номер раздела, в котором описывается эта команда.

## 2.3 Базовый набор команд

### 2.3.1 Входные команды

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Входные команды	LD		Начало операции (загрузка, замыкающий контакт)		*	1	5.1.1
	LDI		Начало операции (загрузка, размыкающий)				
	AND		Последовательное соединение входных контактов				
	ANI		Последовательное соединение входных контактов				
	OR		Параллельное соединение входных контактов				
	ORI		Параллельное соединение входных контактов				
	LDP		Начало операции, управляемой по положит. фронту сигнала		*	2	5.1.2
	LDF		Начало операции, управляемой по отрицат. фронту сигнала				
	ANDP	s	Последоват. соединение по положит. фронту сигнала				
	ANDF	s	Последовательное соединение, по отрицат. фронту сигнала				
ORP	s	Параллельное соединение по положит. фронту сигнала					
ORF	s	Параллельное соединение по отрицат. фронту сигнала					








\*: Количество шагов программы зависит от используемых операндов:

- при использовании внутренних операндов или регистров файлов (с R0 по R32767) : 1
- при использовании непосредственно адресуемых входов (DX) : 2
- при использовании иных операндов : 3
- В случае использования регистров файлов 2R на картах памяти количество шагов может удвоиться.


### 2.3.2 Логические операции

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Логические операции	ANB	—	Последовательное соединение блоков (последовательное соединение параллельных схем)		1	1	5.2.1
	ORB		Параллельное соединение блоков (параллельное соединение последовательных схем)				
	MPS	—	Обработка результата (сохранение результата)		1	1	5.2.2
	MRD		Обработка результата (считывание результата)				
	MPP		Обработка результата (считывание и стирание результата)				
	INV	—	Инвертирование результата операции		1		5.2.3
	MEP	—	Преобразование результатов операции в импульсный вариант (при положительном фронте)		1		5.2.4
	MEF		Преобразование результатов операции в импульсный вариант (при отрицательном фронте)				
	EGP	d	Установка маркера фронта при положительном фронте результата операции		1		5.2.5
	EGF		Установка маркера фронта при отрицательном фронте результата операции				

### 2.3.3 Выходные команды

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Выходные команды	OUT	d	Установка выходов		* 1	* 1	5.3.1
	SET	d	Установка операндов		* 1		5.3.5
	RST	d	Сброс операндов		2	* 1	5.3.6
	PLS	d	Вывод при положительном фронте сигнала		2	* 3	5.3.8
	PLF		Вывод при отрицательном фронте сигнала				
	FF	s	Инвертирование битового выходного операнда		2		5.3.9
	DELTA	d	Выработка импульса переключения на непосредственно адресуемом выходе		2		5.3.11
	DELTAP						

\*: Количество шагов программы зависит от используемых операндов. Точное количество указано в разделах, посвященных отдельным командам.

\*\* : Это  условие выполнения используется только в случае применения маркера ошибки (F).

### 2.3.4 Команды сдвига

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Команды сдвига	SFT	d	Сдвиг битовых операндов		2	* 3	5.4.1
	SFTP						

\*: Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разделе 3.9.2 "Центральные процессоры AnA, AnAS и AnU" этого руководства.



### 2.3.5 Организующие команды

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Организу- ющие команды	MC	n, d	Активация отдельных областей программы		2	* 3/5	5.5.1
	MCR	n	Деактивация отдельных областей программы		1		

\*: Количество шагов равно 5 для команды MC и 3 для команды MCR. Более подробную информацию о количестве шагов в случае применения процессоров AnA, AnAS и AnU см. разделе 3.9.2 "Центральные процессоры AnA, AnAS и AnU" этого руководства.

### 2.3.6 Команды конца программы

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Команды конца программы	FEND	—	Завершение области программы		1		5.6.1
	END		Завершение программы				5.6.2

### 2.3.7 Прочие команды

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Прочие команды	STOP	—	Команда прерывания		1		5.7.1
	NOP	—	Пустой шаг в программе				5.7.2

## 2.4 Прикладные команды I

### 2.4.1 Операторы сравнения

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Сравнение 16-битных данных	LD=	s1, s2	Выход включается, если s1 = s2.		3	* 5/7	6.1.1
	AND=						
	OR=						
	LD<>	s1, s2	Выход включается, если s1 ≠ s2.		3	* 5/7	6.1.1
	AND<>						
	OR<>						
	LD>	s1, s2	Выход включается, если s1 > s2.		3	* 5/7	6.1.1
	AND>						
	OR>						
	LD<=	s1, s2	Выход включается, если s1 ≤ s2.		3	* 5/7	6.1.1
	AND<=						
	OR<=						
	LD<	s1, s2	Выход включается, если s1 < s2.		3	* 5/7	6.1.1
	AND<						
	OR<						
LD>=	s1, s2	Выход включается, если s1 ≥ s2.		3	* 5/7	6.1.1	
AND>=							
OR>=							

\*: Количество шагов программы зависит от используемых операндов.  
Точное количество указано в разделах, посвященных отдельным командам.

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Сравнение 32-битных данных	LDD=	s1, s2	Выход включается, если s1 = s2.		*	**	6.1.2
	ANDD=				3	11	
	ORD=						
	LDD<>	s1, s2	Выход включается, если s1 ≠ s2.		*	**	6.1.2
	ANDD<>				3	11	
	ORD<>						
	LDD>	s1, s2	Выход включается, если s1 > s2.		*	**	6.1.2
	ANDD>				3	11	
	ORD>						
	LDD<=	s1, s2	Выход включается, если s1 ≤ s2.		*	**	6.1.2
	ANDD<=				3	11	
	ORD<=						
	LDD<	s1, s2	Выход включается, если s1 < s2.		*	**	6.1.2
	ANDD<				3	11	
	ORD<						
LDD>=	s1, s2	Выход включается, если s1 ≥ s2.		*	**	6.1.2	
ANDD>=				3	11		
ORD>=							

\*: Количество шагов программы зависит от типа центрального процессора и используемых операндов.

- При использовании процессоров серии QnA и однопроцессорных модулей Q: 3
- При использовании многопроцессорных модулей Q и внутренних словных операндов (кроме регистра файлов ZR): 5  
Константы: 5  
Битовые операнды с адресами, кратными 16,  
имеющие обозначение битового блока K8 и адресуемые без индексации: 5
- При использовании многопроцессорного модуля ЦП серии "Q" и иных операндов кроме вышеуказанных: 3

Хотя в процессоре серии "Q" используется большее число шагов, чем в процессоре QnA, скорость обработки в процессоре серии "Q" выше.



\*\* : Количество шагов программы зависит от используемых операндов.  
Точное количество указано в разделах, посвященных отдельным командам.

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Сравнение чисел с плавающей запятой	LDE=	s1, s2	Выход включается, если s1 = s2.		3		6.1.3
	ANDE=						
	ORE=						
	LDE<>	s1, s2	Выход включается, если s1 ≠ s2.		3		6.1.3
	ANDE<>						
	ORE<>						
	LDE>	s1, s2	Выход включается, если s1 > s2.		3		6.1.3
	ANDE>						
	ORE>						
	LDE<=	s1, s2	Выход включается, если s1 ≤ s2.		3		6.1.3
	ANDE<=						
	ORE<=						
	LDE<	s1, s2	Выход включается, если s1 < s2.		3		6.1.3
	ANDE<						
	ORE<						
LDE>=	s1, s2	Выход включается, если s1 ≥ s2.		3		6.1.3	
ANDE>=							
ORE>=							









Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Сравнение строковых величин	LD $\$$ =	s1, s2	* Строковые величины в s1 и s2 сравниваются знак за знаком. Выход включается, если s1 = s2.		3		6.1.4
	AND $\$$ =						
	OR $\$$ =						
	LD $\$$ <>	s1, s2	* Строковые величины в s1 и s2 сравниваются знак за знаком. Выход включается, если s1 $\neq$ s2.		3		6.1.4
	AND $\$$ <>						
	OR $\$$ <>						
	LD $\$$ >	s1, s2	* Строковые величины в s1 и s2 сравниваются знак за знаком. Выход включается, если s1 > s2.		3		6.1.4
	AND $\$$ >						
	OR $\$$ >						
	LD $\$$ <=	s1, s2	* Строковые величины в s1 и s2 сравниваются знак за знаком. Выход включается, если s1 <= s2.		3		6.1.4
	AND $\$$ <=						
	OR $\$$ <=						
	LD $\$$ <	s1, s2	* Строковые величины в s1 и s2 сравниваются знак за знаком. Выход включается, если s1 < s2.		3		6.1.4
	AND $\$$ <						
	OR $\$$ <						
LD $\$$ >=	s1, s2	* Строковые величины в s1 и s2 сравниваются знак за знаком. Выход включается, если s1 >= s2.		3		6.1.4	
AND $\$$ >=							
OR $\$$ >=							



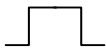

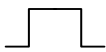

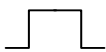

\*: Ниже рассмотрены возможные варианты при сравнении строковых величин.

- Строковые величины идентичны: Все знаки в строковых величинах идентичны.
- Строковая величина большего размера: Если строковые величины различны, определяется величина, содержащая большее количество знаков.
- Строковая величина меньшего размера: Если строковые величины различны, определяется величина, содержащая меньшее количество знаков.

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Поблочное сравнение двоичных данных	ВКСМР=	s1, s2, n, d1	Сравниваются знаки из n 16-битных блоков по адресу s1 со знаками из n 16-битных блоков по адресу s2. Результат сравнения сохраняется в n 16-битных блоках по адресу d1.		5		6.1.5
	ВКСМР<>	s1, s2, n, d1					
	ВКСМР>	s1, s2, n, d1					
	ВКСМР<=	s1, s2, n, d1					
	ВКСМР<	s1, s2, n, d1					
	ВКСМР>=	s1, s2, n, d1					
	ВКСМР=P	s1, s2, n, d1					
	ВКСМР<>P	s1, s2, n, d1					
	ВКСМР>P	s1, s2, n, d1					
	ВКСМР<=P	s1, s2, n, d1					
	ВКСМР<P	s1, s2, n, d1					
	ВКСМР>=P	s1, s2, n, d1					

## 2.4.2 Арифметические команды

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Сложение и вычитание 16-битных двоичных данных	+	s, d	$(d) + (s) \rightarrow (d)$		3	5	6.2.1
	+P						
	+	s1, s2, d1	$(s1) + (s2) \rightarrow (d1)$		4	7	6.2.1
	+P						
	-	s, d	$(d) - (s) \rightarrow (d)$		3	5	6.2.1
	-P						
	-	s1, s2, d1	$(s1) - (s2) \rightarrow (d1)$		4	7	6.2.1
	-P						

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка	
					Q	A		
Сложение и вычитание 32-битных двоичных данных	D+	s, d	$(d + 1, d) + (s + 1, s) \rightarrow (d + 1, d)$		*	3	9	6.2.2
	D + P							
	D+	s1, s2, d1	$((s1) + 1, s1) + ((s2) + 1, s2) \rightarrow ((d1) + 1, d1)$		**	4	11	6.2.2
	D + P							
	D-	s, d	$(d + 1, d) - (s + 1, s) \rightarrow (d + 1, d)$		*	3	9	6.2.2
	D - P							
	D-	s1, s2, d1	$((s1) + 1, s1) - ((s2) + 1, s2) \rightarrow ((d1) + 1, d1)$		**	4	11	6.2.2
	D - P							

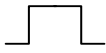

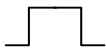

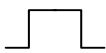

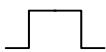

\*: Количество шагов программы зависит от типа центрального процессора и используемых операндов.

- При использовании процессоров серии QnA или однопроцессорного модуля ЦП серии "Q": 3
- При использовании многопроцессорного модуля ЦП серии "Q" и внутренних словных операндов (кроме регистра файлов ZR): 5  
Константы: 5  
Битовые операнды с адресом, кратным 16, имеющие обозначение битового блока K8 и адресуемые без индексации: 5
- При использовании многопроцессорного модуля ЦП серии "Q" и иных операндов кроме вышеуказанных: 3

\*\* : Количество шагов программы зависит от типа центрального процессора и используемых операндов.

- При использовании процессоров серии QnA или однопроцессорного модуля ЦП серии "Q": 4
- При использовании многопроцессорного модуля ЦП серии "Q" и внутренних словных операндов (кроме регистра файлов ZR): 6  
Константы: 6  
Битовые операнды с адресом, кратным 16, имеющие обозначение битового блока K8 и адресуемые без индексации: 6
- При использовании многопроцессорного модуля ЦП серии "Q" и иных операндов кроме вышеуказанных: 4



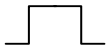

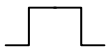

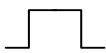

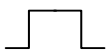

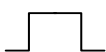







Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка		
					Q	A			
Умножение и деление 16-битных двоичных данных	x	s1, s2, d1	$(s1) \times (s2) \rightarrow ((d1) + 1, d1)$		*	4	**	7	6.2.4
	xP								6.2.4
	/	s1, s2, d1	$(s1)/(s2) \rightarrow$ частное (d1), остаток ((d1) + 1)		*	4	**	7	6.2.4
	/P								6.2.4
Умножение и деление 32-битных двоичных данных	Dx	s1, s2, d1	$((s1) + 1, s1) \times ((s2) + 1, s2)$ $\rightarrow$ $((d1) + 3, (d1) + 2, (d1) + 1, d1)$		*	4	**	11	6.2.4
	DxP								6.2.4
	D/	s1, s2, d1	$((s1) + 1, s1)/((s2) + 1, s2)$ $\rightarrow$ частное ((d1) + 1, d1), остаток ((d1) + 3, (d1) + 2)		*	4	**	11	6.2.4
	D/P								6.2.4

\*: Количество шагов программы зависит от типа центрального процессора и используемых операндов.

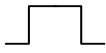

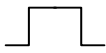

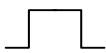

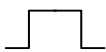

- При использовании процессоров серии QnA или однопроцессорного модуля ЦП серии "Q": 4
- При использовании многопроцессорного модуля ЦП серии "Q" и внутренних словных операндов (кроме регистра файлов ZR): 3  
Константы: 3  
Битовые операнды с адресом, кратным 16, имеющие обозначение битового блока K4 и адресуемые без индексации: 3
- При использовании многопроцессорного модуля ЦП серии "Q" и иных операндов кроме вышеуказанных: 4

\*\* : Количество шагов программы зависит от используемых операндов.

Точное количество указано в разделах, посвященных отдельным командам.

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Сложение и вычитание двоично-десятичных данных (4 знака)	B+	s, d	$(d) + (s) \rightarrow (d)$		3	* 7	6.2.5
	B + P						6.2.5
	B+	s1, s2, d1	$(s1) + (s2) \rightarrow (d1)$		4	* 9	6.2.5
	B + P						6.2.5
	B-	s, d	$(d) - (s) \rightarrow (d)$		3	* 7	6.2.5
	B - P						6.2.5
	B-	s1, s2, d1	$(s1) - (s2) \rightarrow (d1)$		4	* 9	6.2.5
	B - P						6.2.5
Сложение и вычитание двоично-десятичных данных (8 знаков)	DB+	s, d	$(d + 1, d) + (s + 1, s) \rightarrow (d + 1, d)$		3	* 9	6.2.6
	DB + P						6.2.6
	DB+	s1, s2, d1	$((s1) + 1, s1) + ((s2) + 1, s2) \rightarrow ((d1) + 1, d1)$		4	* 11	6.2.6
	DB + P						6.2.6
	DB-	s, d	$(d + 1, d) + (s + 1, s) \rightarrow (d + 1, d)$		3	* 9	6.2.6
	DB - P						6.2.6
	DB-	s1, s2, d1	$((s1) + 1, s1) + ((s2) + 1, s2) \rightarrow ((d1) + 1, d1)$		4	* 11	6.2.6
	DB - P						6.2.6

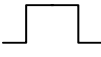
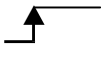
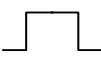

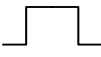
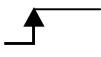
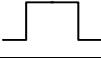

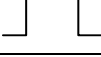



\*: Количество шагов программы зависит от используемых операндов.  
Точное количество указано в разделах, посвященных отдельным командам.

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Умножение и деление двоично-десятичных данных (4 знака)	Vx	s1, s2, d1	$(s1) \times (s2) \rightarrow ((d1) + 1, d1)$		4	* 9	6.2.7
	V × P						
	V/	s1, s2, d1	$(s1)/(s2) \rightarrow$ частное (d1), остаток ((d1) + 1)		4	* 9	6.2.7
	V/P						
Умножение и деление двоично-десятичных данных (8 знаков)	DBx	s1, s2, d1	$((s1)+1, s1) \times ((s2)+1, s2)$ $\rightarrow$ $((d1) + 3, (d1) + 2, (d1) + 1, d1)$		4	* 11	6.2.8
	DB × P						
	DB/	s1, s2, d1	$((s1) + 1, s1) / ((s2) + 1, s2)$ $\rightarrow$ частное ((d1) + 1, d1), остаток ((d1) + 3, (d1) + 2)		4	* 11	6.2.8
	DB/P						

\*: Количество шагов программы зависит от используемых операндов.

Точное количество указано в разделах, посвященных отдельным командам.

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Сложение и вычитание чисел с плавающей запятой	E+	s, d	$(d + 1, d) + (s + 1, s) \rightarrow (d + 1, d)$		3		6.2.9
	E + P						6.2.9
	E+	s1, s2, d1	$((s1) + 1, s1) + ((s2) + 1, s2) \rightarrow ((d1) + 1, d1)$		4		6.2.9
	E + P						6.2.9
	E-	s, d	$(d + 1, d) - (s + 1, s) \rightarrow (d + 1, d)$		3		6.2.9
	E - P						6.2.9
	E-	s1, s2, d1	$((s1) + 1, s1) - ((s2) + 1, s2) \rightarrow ((d1) + 1, d1)$		4		6.2.9
	E - P						6.2.9
Умножение и деление чисел с плавающей запятой	Ex	s1, s2, d1	$((s1) + 1, s1) \times ((s2) + 1, s2) \rightarrow ((d1) + 1, d1)$		4		6.2.10
	Ex P						6.2.10
	E/	s1, s2, d1	$((s1) + 1, s1) / ((s2) + 1, s2) \rightarrow \text{частное } ((d1) + 1, d1)$		4		6.2.10
	E/P						6.2.10
Поблочное сложение и вычитание двоичных данных	BK+	s1, s2, d, n	n-ый 16-битный блок из s1 суммируется с n-ым 16-битным блоком из s2.		5		6.2.11
	BK + P						6.2.11
	BK-	s1, s2, d, n	n-ый 16-битный блок из s2 вычитается из n-ого 16-битного блока из s1.		5		6.2.11
	BK - P						6.2.11

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Объединение строковых величин	\$+	s, d	Строковая величина в s присоединяется к строковой величине в d. Объединенная строковая величина сохраняется в d.		3		6.2.12
	\$+P						
	\$+	s1, s2, d1	Строковая величина в s присоединяется к строковой величине в d. Объединенная строковая величина сохраняется в d.		4		6.2.12
	\$+P						
Приращение двоичных данных	INC	d	(d) + 1 → (d)		2	** 3	6.2.13
	INCP						
	DINC	d	(d + 1, d) + 1 → (d + 1, d)		* 2	** 3	6.2.13
	DINCP						
Отрицательное приращение двоичных данных	DEC	d	(d) - 1 → (d)		2	** 3	6.2.14
	DECP						
	DDEC	d	(d + 1, d) - 1 → (d + 1, d)		* 2	** 3	6.2.14
	DDECP						

\*: Количество шагов программы зависит от типа центрального процессора и используемых операндов.

- При использовании процессоров серии QnA или однопроцессорного модуля ЦП серии "Q": 2
- При использовании многопроцессорного модуля ЦП серии "Q" и внутренних словных операндов (кроме регистра файлов ZR): 3  
Константы: 3  
Битовые операнды с адресом, кратным 16,  
имеющие обозначение битового блока K8 и адресуемые без индексации: 3
- При использовании многопроцессорного модуля ЦП серии "Q" и иных операндов кроме вышеуказанных: 2

\*\* : Количество шагов программы зависит от используемых операндов.

Точное количество указано в разделах, посвященных отдельным командам.

2.4.3 Команды конвертирования

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Преобразование в формат BCD	BCD	s, d	преобразование в формат BCD (s) → (d) ↑ BIN (от 0 до 9999)		3	* 5	6.3.1
	BCDP						
	DBCD	s, d	преобразование в формат BCD (s+1, s) → (d1+1, d1) ↑ BIN (от 0 до 99999999)		3	* 9	6.3.1
	DBCDP						
Преобразование в формат BIN	BIN	s, d	преобразование в формат BIN (s) → (d) ↑ BCD (от 0 до 9999)		3	* 5	6.3.2
	BINP						
	DBIN	s, d	преобразование в формат BIN (s+1, s) → (d1+1, d1) ↑ BCD (от 0 до 99999999)		3	* 9	6.3.2
	DBINP						
Преобразование двоичных чисел в числа с плавающей запятой	FLT	s, d	преобразование в формат BIN (s+1, s) → (d) ↑ число с плавающей запятой (от -32768 до 32767)		3		6.3.3
	FLTP						
	DFLT	s, d	преобразование в формат BIN (s+1, s) → (d1+1, d) ↑ число с плавающей запятой (от -2147483648 до 2147483647)		3		6.3.3
	DFLTP						
Преобразование чисел с плавающей запятой в двоичные числа	INT	s, d	преобразование в число с плавающей запятой (s+1, s) → (d) ↑ двоичное число (от -32768 до 32767)		3		6.3.4
	INTP						
	DINT	s, d	преобразование в число с плавающей запятой (s+1, s) → (d+1, d) ↑ двоичное число (от -2147483648 до 2147483647)		3		6.3.4
	DINTP						

\*: Количество шагов программы зависит от используемых операндов.  
Точное количество указано в разделах, посвященных отдельным командам.


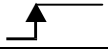







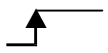
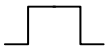

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Преобразование 16-битных двоичных данных в 32-битные двоичные данные	DBL	s, d	преобразование $(s) \rightarrow (d+1, d)$ BIN (от -32768 до 32767)		3		6.3.5
	DBLP						
Преобразование 32-битных двоичных данных в 16-битные двоичные данные	WORD	s, d	преобразование $(s+1, s) \rightarrow (d)$ BIN (от -32768 до 32767)		3		6.3.6
	WORDP						
Преобразование 16- или 32-битных двоичных данных в код Грея	GRY	s, d	преобразование в код Грея $(s) \rightarrow (d)$ двоичное число (от -32768 до 32767)		3		6.3.7
	GRYP						
	DGRY	s, d	преобразование в код Грея $(s+1, s) \rightarrow (d+1, d)$ двоичное число (от -2147483648 до 2147483647)		3		6.3.7
	DGRYP						
Преобразование данных в кодировке Грея в 16/32-битные двоичные данные	GBIN	s, d	преобразование в формат BIN $(s) \rightarrow (d)$ код Грея (от -32768 до 32767)		3		6.3.8
	GBINP						
	DGBIN	s, d	преобразование в формат BIN $(s+1, s) \rightarrow (d+1, d)$ код Грея (от -2147483648 до 2147483647)		3		6.3.8
	DGBINP						
Преобразование 16/32-битных двоичных данных в дополнительный код (изменение арифметического знака)	NEG	d	$(d) \rightarrow (d)$ данные типа BIN		2	* 3	6.3.9
	NEGP						
	DNEG	d	$(d+1, d) \rightarrow (d+1, d)$ данные типа BIN		2	* 3	6.3.9
	DNEGP						

\*: Количество шагов программы зависит от используемых операндов.  
 Точное количество указано в разделах, посвященных отдельным командам.

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Изменение арифметического знака чисел с плавающей запятой	ENEG	d	$(d+1, d)$ → $(d+1, d)$ ↑ число с плавающей запятой		2		6.3.10
	ENEGP						6.3.10
Поблочное преобразование двоичных данных (BIN) в двоично-десятичные (BCD)	BKBCD	s, d, n	Преобразует n-ый 16-битный блок из s и сохраняет преобразованный n-ый 4-разрядный блок двоично-десятичных данных в d.		4		6.3.11
	BKBCDP	s, d, n					6.3.11
Поблочное преобразование двоично-десятичных данных (BCD) в двоичные (BIN)	BKBIN	s, d, n	Преобразует n-ый 4-разрядный блок двоично-десятичных данных из s и сохраняет преобразованный n-ый 16-битный блок в d.		4		6.3.12
	BKBINP	s, d, n					6.3.12



## 2.4.4 Команды передачи

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Передача 16-битных данных	MOV	s, d	(s) → (d)		* 3	*** 5	6.4.1
	MOVP	s, d					
Передача 32-ных данных	DMOV	s, d	(s+1, s) → (d+1, d)		** 3	*** 7	6.4.1
	DMOVP	s, d					
Передача чисел с плавающей запятой	EMOV	s, d	(s+1, s) → (d+1, d1) ↓ число с плавающей запятой		3		6.4.2
	EMOVP	s, d					
Передача строковых величин	\$MOV	s, d	Передает находящиеся в s байты строковой величины в d.		3		6.4.3
	\$MOVP	s, d					
Инверсия 16-битных двоичных данных	CML	s, d	$\overline{(s)} \rightarrow (d)$		* 3	*** 5	6.4.4
	CMLP	s, d					
Инверсия 32-битных двоичных данных	DCML	s, d	$\overline{(s+1, s)} \rightarrow (d1+1, d1)$		** 3	*** 7	6.4.4
	DCMLP	s, d					

\*: Количество шагов программы зависит от типа центрального процессора и используемых операндов.

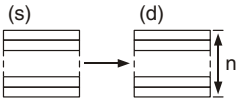

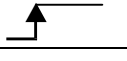
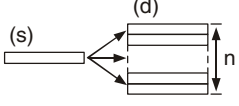
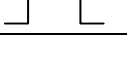
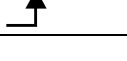
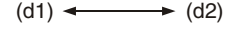
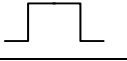

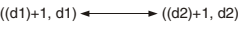
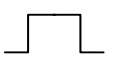

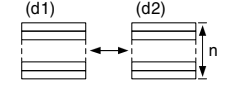
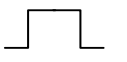
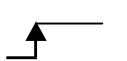
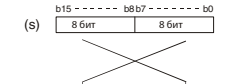


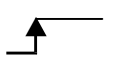
- При использовании процессоров серии QnA или однопроцессорного модуля ЦП серии "Q": 3
- При использовании многопроцессорного модуля ЦП серии "Q" и внутренних словных операндов (кроме регистра файлов ZR): 2  
Константы: 2  
Битовые операнды с адресом, кратным 16, имеющие обозначение битового блока K4 и адресуемые без индексации: 2
- При использовании многопроцессорного модуля ЦП серии "Q" и иных операндов кроме вышеуказанных: 3

\*\* : Количество шагов программы зависит от типа центрального процессора:

- При использовании однопроцессорного модуля ЦП серии "Q": 2
- При использовании процессоров QnA или многопроцессорного модуля ЦП серии "Q": 3

\*\*\*: Количество шагов программы зависит от используемых операндов.

Точное количество указано в разделах, посвященных отдельным командам.

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
Передача блоков двоичных данных	BMOV	s, n, d			4	* 9	6.4.5
	BMOVP	s, n, d					
Передача блока двоичных данных	FMOV	s, n, d			4	* 9	6.4.6
	FMOVP	s, n, d					
Перестановка 16-битных двоичных данных	XCH	d1, d2			3	* 5	6.4.7
	XCHP	d1, d2					
Перестановка 32-битных двоичных данных	DXCH	d1, d2			3	* 7	6.4.7
	DXCHP	d1, d2					
Перестановка блоков двоичных данных	BXCH	n, d1, d2			4		6.4.8
	BXCHP	n, d1, d2					
Перестановка байтов в пределах двоичного числа	SWAP	s			3		6.4.9
	SWAPP	s					

\*: Количество шагов программы зависит от используемых операндов.  
Точное количество указано в разделах, посвященных отдельным командам.

### 2.4.5 Команды разветвления программы

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Команды переходов	CJ	p	Условный переход (p = цель перехода)		2	* 3	6.5.1
	SCJ	p	Условный переход в следующем цикле (p = цель перехода)				
	JMP	p	Команда перехода (p = цель перехода)		2	* 3	6.5.1
	GOEND		Переход в конец программы		1		6.5.2


\*: Количество шагов программы зависит от используемых операндов.  
Точное количество указано в разделах, посвященных отдельным командам.

### 2.4.6 Команды для вызова программ прерывания

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Прерывание запрещено	DI		Предотвращает выполнение программы прерывания		1	* 1	6.6.1
Прерывание разрешено	EI		Разрешает вызов программы прерывания		1	* 1	6.6.1
Битовая схема условий выполнения программ прерывания	IMASK	s	Указанная в s битовая схема содержит биты, сопоставленные отдельным адресам прерывания.		2	* 1	6.6.1
Возврат из программы прерывания в основную программу	IRET		Конец программы прерывания		1	* 1	6.6.2

\*: Количество шагов программы зависит от используемых операндов.  
Точное количество указано в разделах, посвященных отдельным командам.


### 2.4.7 Команда обновления данных

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Частичное обновление ввода-вывода	RFS	s, n	Команда RFS обновляет входы и выходы выбранной области на один программный цикл.		3		6.7.1
Частичное обновление ввода-вывода	SEG	s, d	Команда SEG позволяет обновлять определенную область адресов ввода-вывода при выполненном входном условии.			* 9	6.7.2
Команда обновления сетевых и интерфейсных данных	COM		Если SM775 (у MELSEC Q) не установлен (0), выполняется обновление сетевых и интерфейсных данных (обновление канала связи) и обработка всех данных (обработка команды END).		1	* 3	6.7.3
Предотвращение обновления канала связи	DI		Команда DI предотвращает обновление канала связи до тех пор, пока в очереди команд программы не появится команда EI.		1		6.7.4
Разрешение обновления канала связи	EI		После установки команды EI обновление канала возможно.		1		6.7.4

\*: Количество шагов программы зависит от используемых операндов.  
Точное количество указано в разделах, посвященных отдельным командам.

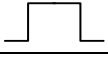
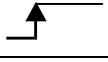
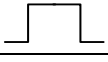
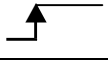
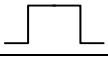

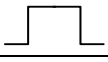

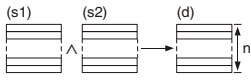
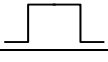

## 2.4.8 Прочие команды

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Однофазный возрастающий/убывающий счетчик	UDCNT1	s, n, d	<p>текущее значение счетчика момент переключения контакта счетчика</p>		4		6.8.1
Двухфазный возрастающий/убывающий счетчик	UDCNT2	s, n, d	<p>текущее значение счетчика момент переключения контакта счетчика</p>		4		6.8.2
Программируемый таймер	TTMR	d, n	(время, в течение которого таймер TTMR установлен) x n → (d) n = 0:1, n = 1:10, n = 2:100		3		6.8.3
Таймер для специальных функций (медленный таймер)	STMR	s, n, d	Команда STMR использует выходы, указанные по адресу с d + 0 по d + 3, для четырех различных функций таймера: d + 0: выход таймера с задержкой выключения d + 1: выход таймера с устанавливающим импульсом (установка при отриц. фронте) d + 2: выход таймера с устанавливающим импульсом (установка при положит. фронте) d + 3: выход таймера с задержкой включения		3		6.8.4
Таймер для специальных функций (быстрый таймер)	STMRH	s, n, d	см. выше		3		6.8.4
Команда позиционирования поворотных столов	ROTC	s, n1, n2, d	С помощью команды ROTC поворотный стол с количеством секторов (делений), указанным в n1, поворачивается таким образом, чтобы сектор, указанный в s + 2, переместился в позицию, указанную в s + 1.		5		6.8.5
Линейно изменяющийся сигнал	RAMP	n1, n2, n3, d1, d2	Команда RAMP плавно повышает содержимое (d1) + 0 с начального значения, указанного в n1, до конечного значения, указанного в n2.		6		6.8.6
Счетчик импульсов	SPD	s, n, d	Команда SPD считает импульсы указанного в s входа в течение указ. в n времени и сохраняет результат измерения в d.		4		6.8.7
Импульсный выход с настраиваемым количеством импульсов	PLSY	s1, s2, d	Команда PLSY выводит количество импульсов, указанное в s2, с частотой, указанной в s1, на выход, указанный в d.		4		6.8.8
Широтно-импульсная модуляция	PWM	n1, n2, d			4		6.8.9

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Образование матрицы ввода	MTR	s, n, d1, d2	Команда MTR считывает 16 битов информации, начиная с s. Количество повторений этого процесса (число рядов) указано в n. Состояния считанной информации сохраняются, начиная с d2.		5		6.8.10

## 2.5 Прикладные команды, часть II

### 2.5.1 Логические команды

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Логическое произведение	WAND	s, d	$(d) \wedge (s) \longrightarrow (d)$		3	*** 5	7.1.1
	WANDP						
	WAND	s1, s2, d1	$(s1) \wedge (s2) \longrightarrow (d1)$		4	*** 7	7.1.1
	WANDP						
	DAND	s, d	$(d + 1, d) \wedge (s + 1, s) \longrightarrow (d + 1, d)$		* 4	*** 9	7.1.1
	DANDP						
	DAND	s1, s2, d	$((s1) + 1, s1) \wedge ((s2) + 1, s2) \longrightarrow (d + 1, d)$		** 4		7.1.1
	DANDP						
BKAND	s1, s2, n, d			5		7.1.2	
BKANDP							

\*: Количество шагов программы зависит от типа центрального процессора и используемых операндов.

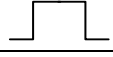

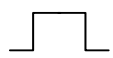

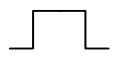

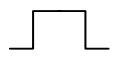
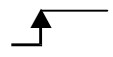
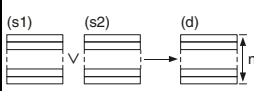


- При использовании процессоров QnA: 4
- При использовании однопроцессорного модуля ЦП серии "Q": 3
- При использовании многопроцессорного модуля ЦП серии "Q" и внутренних словных операндов (кроме регистра файлов ZR): 6  
Константы: 6  
Битовые операнды с адресом, кратным 16, имеющие обозначение битового блока K8 и адресуемые без индексации: 6
- При использовании многопроцессорного модуля ЦП серии "Q" и иных операндов кроме вышеуказанных: 4

\*\* : Количество шагов программы зависит от типа центрального процессора и используемых операндов.

- При использовании процессоров QnA: 4
- При использовании процессора "System Q" и внутренних словных операндов (кроме регистра файлов ZR): 6  
Константы: 6  
Битовые операнды с адресом, кратным 16, имеющие обозначение битового блока K8 и адресуемые без индексации: 6
- При использовании процессора "System Q" и иных операндов кроме вышеуказанных: 4

\*\*\*: Количество шагов программы зависит от используемых операндов.

Точное количество указано в разделах, посвященных отдельным командам.

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Логическая сумма	WOR	s, d	$(d) \vee (s) \longrightarrow (d)$		3	*** 5	7.1.3
	WORP						
	WOR	s1, s2, d1	$(s1) \vee (s2) \longrightarrow (d1)$		4	*** 7	7.1.3
	WORP						
	DOR	s, d	$(d + 1, d) \vee (s + 1, s) \longrightarrow (d + 1, d)$		* 4	*** 9	7.1.3
	DORP						
	DOR	s1, s2, d	$((s1) + 1, s1) \vee ((s2) + 1, s2) \longrightarrow (d + 1, d)$		** 4		7.1.3
	DORP						
BKOR	s1, s2, n, d			5		7.1.4	
BKORP							

\*: Количество шагов программы зависит от типа центрального процессора и используемых операндов.

- При использовании процессоров QnA: 4
  - При использовании однопроцессорного модуля ЦП серии "Q": 3
  - При использовании многопроцессорного модуля ЦП серии "Q" и внутренних словных операндов (кроме регистра файлов ZR): 6
- Константы: 6  
 Битовые операнды с адресом, кратным 16, имеющие обозначение битового блока K8 и адресуемые без индексации: 6
- При использовании многопроцессорного модуля ЦП серии "Q" и иных операндов кроме вышеуказанных: 4

\*\* : Количество шагов программы зависит от типа центрального процессора и используемых операндов.

- При использовании процессоров QnA: 4
  - При использовании процессора "System Q" и внутренних словных операндов (кроме регистра файлов ZR): 6
- Константы: 6  
 Битовые операнды с адресом, кратным 16, имеющие обозначение битового блока K8 и адресуемые без индексации: 6
- При использовании процессора "System Q" и иных операндов кроме вышеуказанных: 4

\*\*\*: Количество шагов программы зависит от используемых операндов.

Точное количество указано в разделах, посвященных отдельным командам.



Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Логика "исключающее ИЛИ"	WXOR	s, d	$(d) \vee (s) \longrightarrow (d)$		3	*** 5	7.1.5
	WXORP						
	WXOR	s1, s2, d1	$(s1) \vee (s2) \longrightarrow (d1)$		4	*** 7	7.1.5
	WXORP						
	DXOR	s, d	$(d + 1, d) \vee (s + 1, s) \longrightarrow (d + 1, d)$		* 4	*** 9	7.1.5
	DXORP						
	DXOR	s1, s2, d	$((s1) + 1, s1) \vee ((s2) + 1, s2) \longrightarrow (d + 1, d)$		** 4		7.1.5
	DXORP						
BKXOR	s1, s2, n, d			5		7.1.6	
BKXORP							

\*: Количество шагов программы зависит от типа центрального процессора и используемых операндов.

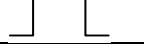
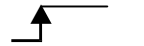

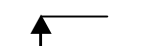

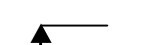
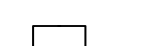
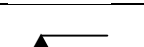
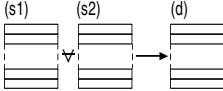
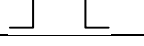

- При использовании процессоров QnA: 4
- При использовании однопроцессорного модуля ЦП серии "Q": 3
- При использовании многопроцессорного модуля ЦП серии "Q" и внутренних словных операндов (кроме регистра файлов ZR): 6  
Константы: 6  
Битовые операнды с адресом, кратным 16, имеющие обозначение битового блока K8 и адресуемые без индексации: 6
- При использовании многопроцессорного модуля ЦП серии "Q" и иных операндов кроме вышеуказанных: 4

\*\* : Количество шагов программы зависит от типа центрального процессора и используемых операндов.

- При использовании процессоров QnA: 4
- При использовании процессора Q и внутренних словных операндов (кроме регистра файлов ZR): 6  
Константы: 6  
Битовые операнды с адресом, кратным 16, имеющие обозначение битового блока K8 и адресуемые без индексации: 6
- При использовании процессора Q и иных операндов кроме вышеуказанных: 4

\*\*\*: Количество шагов программы зависит от используемых операндов.

Точное количество указано в разделах, посвященных отдельным командам.

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Логика "исключающее НЕ ИЛИ"	WNXR	s, d	$(d) \vee (s) \longrightarrow (d)$		3	*** 5	7.1.7
	WNXRP						
	WNXR	s1, s2, d1	$(s1) \nabla (s2) \rightarrow (d1)$		4	*** 7	7.1.7
	WNXRP						
	DNXR	s, d	$(d+1, d) \nabla (s+1, s) \rightarrow (d+1, d)$		*	*** 9	7.1.7
	DNXRP						
	DNXR	s1, s2, d	$((s1)+1, s1) \nabla ((s2)+1, s2) \rightarrow (d+1, d)$		**	4	7.1.7
	DNXRP						
	BKXNR	s1, s2, n, d			5		7.1.8
	BKXNRP						

\*: Количество шагов программы зависит от типа центрального процессора и используемых операндов.

- При использовании процессоров QnA: 4
- При использовании однопроцессорного модуля ЦП серии "Q": 3
- При использовании многопроцессорного модуля ЦП серии "Q" и внутренних словных операндов (кроме регистра файлов ZR): 6  
Константы: 6  
Битовые операнды с адресом, кратным 16, имеющие обозначение битового блока K8 и адресуемые без индексации: 6
- При использовании многопроцессорного модуля ЦП серии "Q" и иных операндов кроме вышеуказанных: 4

\*\* : Количество шагов программы зависит от типа центрального процессора и используемых операндов.

- При использовании процессоров QnA: 4
- При использовании процессора Q и внутренних словных операндов (кроме регистра файлов ZR): 6  
Константы: 6  
Битовые операнды с адресом, кратным 16, имеющие обозначение битового блока K8 и адресуемые без индексации: 6
- При использовании процессора Q и иных операндов кроме вышеуказанных: 4

\*\*\*: Количество шагов программы зависит от используемых операндов.

Точное количество указано в разделах, посвященных отдельным командам.

## 2.5.2 Команды вращения

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Вращение данных вправо (16 бит)	ROR	n, d			3	* 3	7.2.1
	RORP		вращение на n битов вправо				
	RRCR	n, d			3	* 3	7.2.1
	RRCRP		вращение на n битов вправо				
Вращение данных влево (16 бит)	ROL	n, d			3	* 3	7.2.2
	ROLP		вращение на n битов влево				
	RCL	n, d			3	* 3	7.2.2
	RCLP		вращение на n битов влево				
Вращение данных вправо (32 бита)	DROR	n, d			3	* 3	7.2.3
	DRORP		вращение на n битов вправо				
	DRCR	n, d			3	* 3	7.2.3
	DRCRP		вращение на n битов вправо				
Вращение данных влево (32 бита)	DROL	n, d			3	* 3	7.2.4
	DROLP		вращение на n битов влево				
	DRCL	n, d			3	* 3	7.2.4
	DRCLP		вращение на n битов влево				

\*: Количество шагов программы зависит от используемых операндов.

Точное количество указано в разделах, посвященных отдельным командам.

2.5.3 Команды сдвига

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Сдвиг 16-битного слова данных на n битов	SFR	n, d			3	* 3	7.3.1
	SFRP						
	SFL	n, d			3	* 3	7.3.1
	SFLP						
Сдвиг n-битного операнда на 1 бит	BSFR	n, d			3	* 7	7.3.2
	BSFRP						
	BSFL	n, d			3	* 7	7.3.2
	BSFLP						
Сдвиг n-словных операндов на 1 адрес	DSFR	n, d			3	* 7	7.3.3
	DSFRP						
	DSFL	n, d			3	* 7	7.3.3
	DSFLP						

\*: Количество шагов программы зависит от используемых операндов.  
Точное количество указано в разделах, посвященных отдельным командам.

### 2.5.4 Команды обработки битов

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Установка/ сброс отдельных битов	BSET	n, d	(d) b15    bn    b0 		3	* 3	7.4.1
	BSETP						
	BRST	n, d	(d) b15    bn    b0 		3	* 7	
	BRSTP						
Опрос состояния отдельных битов в 16/32-битных словах данных	TEST	s1, s2, d	(s1) b15    до    b0    (d) 		4		7.4.2
	TESTP						
	DTEST	s1, s2, d	(s1) b31    до    b0    (d) 		4		
	DTESTP						
Сброс областей битов	BKRST	s, n	(s) ВКЛ. Выкл. RESET Выкл. ВКЛ. 		3		7.4.3
	BKRSTP						

\*: Количество шагов программы зависит от используемых операндов.  
Точное количество указано в разделах, посвященных отдельным командам.


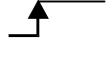
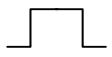



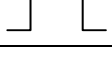
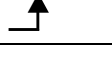
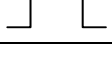

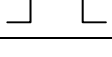
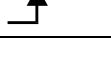
### 2.5.5 Команды обработки данных

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Поиск 16-битных данных	SER	s1, s2, n (A) s1, s2, n, d (Q)			5	* 9	7.5.1
	SERP						
Поиск 32-битных данных	DSER	s1, s2, n (A) s1, s2, n, d (Q)			5	* 9	7.5.1
	DSERP						
Контроль битов данных (16/32 бита)	SUM	s (A) s, d (Q)			3	* 3	7.5.2
	SUMP						
	DSUM	s (A) s, d (Q)			3	* 3	7.5.2
	DSUMP						
Декодировка данных	DECO	s, d, n	<p>Декодировка с 8 на 256 бит</p>		4	* 9	7.5.3
	DECOP						
Кодировка данных	ENCO	s, d, n	<p>Кодировка с 256 на 8 бит</p>		4	* 9	7.5.4
	ENCOP						
7-сегментная декодировка	SEG	s, d			3	7	7.5.5
	SEGP						

\*: Количество шагов программы зависит от используемых операндов.  
Точное количество указано в разделах, посвященных отдельным командам.


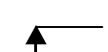
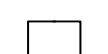

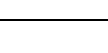

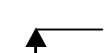
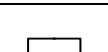
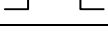

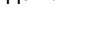
Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Разъединение и объединение 16-битных слов данных	DIS	s, n, d	16-битные значения данных разделяются на группы по 4 бита. Разделяемые значения данных указаны в s, количество 4-битных групп в n, а первый целевой адрес – в d.		4	* 9	7.5.6
	DISP						
	UNI	s, n, d	Отделяются по 4 младших бита 16-битных значений данных, количество которых может достигать четырех, и их состояния объединяются в одно 16-битное значение данных.		4	* 9	7.5.7
	UNIP						
	NDIS	s1, s2, d	Данные, начиная с s1, разделяются на битовые группы, размер которых указан в s2. Разделенные битовые группы сохраняются по отдельности, начиная с d.		4		7.5.8
	NDISP						
	NUNI	s1, s2, d	Группы битов, размер которых указан начиная с s2, выделяются из s1 и объединяются в одно значение данных. Группы битов сохраняются подряд, начиная с d.		4		7.5.9
	NUNIP						
	WTOB	s, n, d	В этой команде разделяемые значения данных указываются в s, количество групп байтов в n, а первый целевой адрес – в d.		4		7.5.9
	WTOBP						
	BTOW	s, n, d	Начальный адрес объединяемых значений данных устанавливается в s, количество считываемых подряд групп байтов в n, а целевой адрес – в d.		4		7.5.10
	BTOWP						
Поиск максимальных значений в 16/32-битных данных	MAX	s, n, d	Поиск максимального значения в 16-битных блоках данных. Количество блоков данных, в которых ведется поиск, указывается в n. Максимальное значение, найденное в данных от s до s + (n - 1), сохраняется в d.		4		7.5.10
	MAXP						
	DMAX	s, n, d	Поиск максимального значения в 32-битных блоках данных. Количество блоков данных, в которых ведется поиск, указывается в n. Максимальное значение, найденное в данных от s до s + (n - 1), сохраняется в d.		4		7.5.10
	DMAXP						

\*: Количество шагов программы зависит от используемых операндов.  
Точное количество указано в разделах, посвященных отдельным командам.

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка		
					Q	A			
Поиск минимальных значений в 16/32-битных данных	MIN	s, n, d	Поиск минимального значения в 16-битных блоках данных. Количество блоков данных, в которых ведется поиск, указывается в n. Минимальное значение, найденное в данных от s до s + (n - 1), сохраняется в d.		4		7.5.11		
	MINP								
	DMIN	s, n, d	Поиск минимального значения в 32-битных блоках данных. Количество блоков данных, в которых ведется поиск, указывается в n. Минимальное значение, найденное в данных от s до s + (n - 1), сохраняется в d.						
	DMINP								
Сортировка 16/32-битных данных	SORT	s1, n, s2, d1, d2	Сортирует указанные в s1 16-битные данные, количество которых указано в n, в возрастающей или убывающей последовательности.		6		7.5.12		
	SORTP								
	DSORT		Сортирует указанные в s1 32-битные данные, количество которых указано в n, в возрастающей или убывающей последовательности.						
	DSORTP								
Суммирование блоков 16/32-битных двоичных данных	WSUM	s, n, d	Суммирует n 16-битных двоичных значений, начиная с s, и сохраняет результат в d и d + 1.		4		7.5.13		
	WSUMP								
	DWSUM	s, n, d	Суммирует n 32-битных двоичных значений, начиная с s и s + 1, и сохраняет результат в d...d + 3.				4		7.5.14
	DWSUMP								


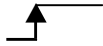


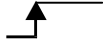


## 2.5.6 Команды структурирования программы

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Команды повторения	FOR	n	Комбинация команд FOR-NEXT позволяет повторять отдельные фрагменты программы без задания входного условия. Количество повторений устанавливается в n.		2	* 3	7.6.1
	NEXT				1	* 1	
	BREAK	p, d	Команда BREAK прерывает цикл FOR-NEXT во время его выполнения и осуществляет переход к указателю (метке), указанному в p.		3		7.6.2
	BREAKP						
Вызов подпрограммы	CALL	p	С помощью команды CALL вызывается подпрограмма путем указания ее метки.		2	* 3	7.6.3
	CALLP						
	RET		Командой RET обозначается конец подпрограммы.		1	* 1	7.6.4
	FCALL	p	При сбросе условия выполнения команды FCALL контакты и катушки подпрограммы, указанной в p (указатель/метка), переводятся в состояние, которое они имели бы, если бы условия выполнения соответствующих команд не были установлены.		2		7.6.5
	FCALLP						
Вызов подпрограммы с обращением к другому программному файлу	ECALL	название файла, p	Для вызова подпрограммы с помощью команды ECALL указывается название программного файла и метка подпрограммы.		3		7.6.6
	ECALLP						

\*: Количество шагов программы зависит от используемых операндов.

Точное количество указано в разделах, посвященных отдельным командам.

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Вызов подпрограммы с обращением к другому программному файлу	EFCALL	название файла, p	При сбросе условия выполнения команды EFCALL контакты и катушки подпрограммы, указанной в p (указатель/метка), в программном файле с указанным названием, переводятся в состояние, которое они имели бы, если бы условия выполнения команд, управляющих контактам и катушками, не были установлены.		3		7.6.7
	EFCALLP						
Переключение между главной программой и подпрограммой	CHG		При выполненном входном условии эта команда осуществляет переключение между основной программой и подпрограммой.			1	7.6.8
Вызов программы микрокомпьютера	SUB	n	При выполненном входном условии команда SUB вызывает программу микрокомпьютера, находящуюся по адресу n.		3		7.6.9
	SUBP						
Индексированная адресация части программы	IX	s	Команды IX и IXEND выполняют индексацию адресов операндов в той части программы, которая находится между командами IX и IXEND.		2		7.6.10
	IXEND				1		
Сохранение индексированных адресов операндов в индексном перечне	IXDEV	p, d	Команды IXDEV и IXSET считывают адреса операндов, расположенные в области смещения, и записывают эти значения смещения в виде индексного перечня в операнд, указанный в d.		1		7.6.11
	IXSET				3		

### 2.5.7 Команды для работы с перечнями данных

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Запись данных в перечень	FIFW	s, d			3	* 7	7.7.1
	FIFWP						
Считывание данных, введенных первыми, из перечня данных	FIFR	s, d			3	* 7	7.7.2
	FIFRP						
Считывание данных, введенных последними, из перечня данных	FPOP	s, d			3		7.7.3
	FPOPP						
Стирание определенных блоков данных в перечне данных	FDEL	s, n, d			4		7.7.4
	FDELP						
Ввод определенных блоков данных в перечень данных	FINS						
	FINSP						








\*: Количество шагов программы зависит от используемых операндов.  
Точное количество указано в разделах, посвященных отдельным командам.

2.5.8 Команды обращения к буферной памяти

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Считывание данных из специального модуля	FROM	n1, n2, n3, d	Считывает 1-словные данные (16 бит) из буферной памяти специального модуля.		5	* 9	7.8.1
	FROMP						
	DFRO		Считывает 2-словные данные (32 бита) из буферной памяти специального модуля.				
	DFROP						
Запись данных в специальный модуль	TO	s, n1, n2, n3	Записывает 1-словные данные (16 бит) из памяти центрального процессора в буферную память специального модуля.		5	* 9	7.8.2
	TOP						
	DTO		Записывает 2-словные данные (32 бита) в буферную память специального модуля.				
	DTOP						
Считывание данных из удаленной станции	FROM, PRC	n1, n2, n3, d (FROM(P)/DFRO(P)) s, d PRC	Считывает 1-словные данные (16 бит) из удаленной станции			* 7/9	7.8.3
	FROMP, PRC						
	DFRO, PRC		Считывает 2-словные данные (32 бита) из удаленной станции				
	DFROP, PRC						
Запись данных в удаленную станцию	TO, PRC	s, n1, n2, n3 (TO(P)/DTO(P)) s, d (PRC)	Записывает 1-словные данные (16 бит) в удаленную станцию			* 7/9	7.8.4
	TOP, PRC						
	DTO, PRC		Записывает 2-словные данные (32 бита) в удаленную станцию				
	DTOP, PRC						

\*: Количество шагов программы зависит от используемых операндов.  
Точное количество указано в разделах, посвященных отдельным командам.



## 2.5.9 Команды индикации




Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Вывод в кодировке ASCII	PR	s, d	Маркер SM701 (серии "Q"/"System Q") установлен (1): Вывод на выходной модуль строковой величины в кодировке ASCII длиной в 16 знаков. Эта строковая величина, распределенная на 2 группы по 8 знаков, считывается из области адресов s и выводится на выходы, указанные в d. Маркер SM701 (серии "Q"/"System Q") не установлен(0): Выводится строковая величина в виде шестнадцатеричных кодов ASCII до кода "00н" из области адресов, указанной в s, на выходы, указанные в d.		3	* 7	7.9.1
	PRC	s, d	Выводит на выходной модуль комментарий (в ASCII-кодировке). Если маркер SM701 (серии "Q"/"System Q") установлен (1), выводятся 16 знаков. Если маркер SM701 не установлен (0), выводятся 32 знака.		3	* 7	7.9.2
Индикация ASCII-знаков и комментариев	LED	s	Команда LED вызывает ASCII-данные (16 знаков) из 8 адресов заданной области и отображает их на светодиодном дисплее процессорного модуля, имеющего соответствующее оснащение.		2	* 3	7.9.3
	LEDC	s (Q)	Команда LEDC вызывает комментарии (16 знаков) из заданной области адресов (s) и отображает их на светодиодном дисплее процессорного модуля, имеющего соответствующее оснащение.		2	* 3	7.9.4
	LEDA	n	Индикация ASCII-знаков на светодиодном дисплее центрального процессора			* 13	7.9.5
	LEDB						
Стирание индикации	LEDR		Сброс маркеров ошибки и индикации на дисплее		1	* 1	7.9.6

\*: Количество шагов программы зависит от используемых операндов.

Точное количество указано в разделах, посвященных отдельным командам.

2.5.10 Распознавание и устранение ошибок




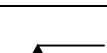
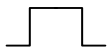
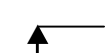
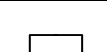
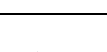






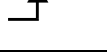
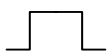
Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Контроль на ошибки	CHKST		Запускает выполнение команды CHK. Если условие выполнения команды CHKST не установлено (0), то выполняется очередной шаг программы, следующий за командой CHK.		1		7.10.1
	CHK (Q)		Проверяет на наличие ошибок схему контактов с концевыми выключателями. Если в схеме имеется ошибка, то устанавливается операнд d1 и соответствующий код ошибки записывается в операнд d2.				
	CHK (A)	d1, d2	Проверяет на наличие ошибок схему контактов с концевыми выключателями. Если в схеме имеется ошибка, то устанавливается операнд d1 и соответствующий код ошибки записывается в операнд d2.		5		7.10.2
	CHKCIR		Вырабатывает контрольные схемы для команды CHK и запускает область программы с этими контрольными схемами.		1		7.10.3
	CHKEND		Команда End для области программы, содержащей контрольные схемы.				
Фиксация и сброс фиксации состояния	SLT		Выполняет промежуточное сохранение в памяти (фиксацию) определенных данных операндов. Эти данные фиксируются в т. н. памяти-защелке, где они могут быть проверены и выведены на индикацию.		1	1	7.10.4
	SLTR		Стирает данные, промежуточно сохраненные в области фиксации. Команда SLT сбрасывается.				
Установка/сброс семплирования (выборочной трассировки)	STRA		Установка выборочной трассировки		1	1	7.10.5
	STRAR		Сброс выборочной трассировки				









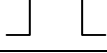
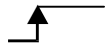



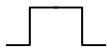



Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Выполнение/ установка/ сброс трассировки программы	PTRA		Установка трассировки программы		1		7.10.6
	PTRAR		Сброс трассировки программы				
	PTRAEXE		Выполнение трассировки программы				
	PTRAEXEP						


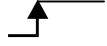







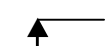
2.5.11 Команды для обработки строковых величин

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Преобразование 16/32-битных двоичных данных в десятичные числа в кодировке ASCII	BINDA	s, d	Преобразует указанное в s 16-битное двоичное число в 5-значное десятичное число в кодировке ASCII и сохраняет его в d.		3		7.11.1
	BINDAP						
	DBINDA		Преобразует указанные в s 32-битные двоичные данные в 10-значное десятичное число в кодировке ASCII и сохраняет его в d.				
	DBINDAP						
Преобразование 16/32-битных двоичных данных в шестнадцатеричные числа в кодировке ASCII	BINHA	s, d	Преобразует указанные в s 16-битные двоичные данные в 4-значное шестнадцатеричное число в кодировке ASCII и сохраняет его в d.		3		7.11.2
	BINHAP						
	DBINHA		Преобразует указанные в s 32-битные двоичные данные в 8-значное шестнадцатеричное число в кодировке ASCII и сохраняет его в d.				
	DBINHAP						
Преобразование 4-/8-значных двоично-десятичных данных в кодировку ASCII	BCDDA	s, d	Преобразует указанные в s 4-значные двоично-десятичные данные в формат ASCII и сохраняет их в d.		3		7.11.3
	BCDDAP						
	DBCDDA		Преобразует указанные в s 8-значные двоично-десятичные данные в формат ASCII и сохраняет их в d.				
	DBCDDAP						
Преобразование десятичных данных ASCII в 16/32-битные двоичные данные	DABIN	s, d	Преобразует указанные в s 5-разрядные десятичные данные в кодировке ASCII в 16-битный двоичный формат и сохраняет их в d.		3		7.11.4
	DABINP						
	DDABIN		Преобразует указанные в s 10-разрядные десятичные данные в кодировке ASCII в 32-битный двоичный формат и сохраняет их в d.				
	DDABINP						



Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Преобразование данных из шестнадцатеричных кодов ASCII в 16/32-битный двоичный формат	HABIN	s, d	Преобразует 4-значные данные в виде шестнадцатеричных кодов ASCII, указанные в s, в 16-битный двоичный формат и сохраняет их в d.		3		7.11.5
	HABINP						
	DHABIN		Преобразует 8-значные данные в виде шестнадцатеричных кодов ASCII, указанные в s, в 32-битный двоичный формат и сохраняет их в d.				
	DHABINP						
Преобразование десятичных данных из кодировки ASCII в 4-/8-значную двоично-десятичную кодировку	DABCD	s, d	Преобразует десятичные данные в кодировке ASCII, указанные в s, в 4-значный формат BCD и сохраняет их в d.		3		7.11.6
	DABCDP						
	DDABCD		Преобразует десятичные данные в кодировке ASCII, указанные в s, в 8-значный формат BCD и сохраняет их в d.				
	DDABCDP						
Считывание данных комментариев	COMRD	s, d	Считывает данные комментариев из области, указанной в s, и сохраняет их в ASCII-кодировке в d.		3		7.11.7
	COMRDP						
Определение длины строковых величин	LEN	s, d	Определяет длину строковой величины, указанной в s, и сохраняет результат в d.		3		7.11.8
	LENP						
Преобразование 16/32-битных двоичных данных в строковые величины	STR	s1, s2, d	Вставляет десятичную запятую (в виде точки) в 16-битное двоичное значение, указанное в s2. Место вставки запятой указано в s1. Результат преобразуется в строковую величину и сохраняется в d.		4		7.11.9
	STRP						
	DSTR		Вставляет десятичную запятую (в виде точки) в 32-битное двоичное значение, указанное в s2. Место вставки запятой указано в s1. Результат преобразуется в строковую величину и сохраняется в d.				
	DSTRP						

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Преобразование строковых величин в 16/32-битные двоичные данные	VAL	s, d1, d2	Преобразует строковые величины, сохраненные в s, в 16-битные двоичные данные. Количество разрядов и двоичное значение сохраняются, начиная с d1 и d2.		4		7.11.10
	VALP						
	DVAL		Преобразует строковые величины, сохраненные в s, в 32-битные двоичные данные. Количество разрядов и двоичное значение сохраняются в d1 и d2.				
	DVALP						
Преобразование чисел с плавающей запятой в строковые величины	ESTR	s1, s2, d	Преобразует числа с плавающей запятой (данные типа REAL), хранящиеся в s1, в строковую величину. Формат этой строковой величины указывается в s2. Результат сохраняется в d.		4		7.11.11
	ESTRP						
Преобразование строковых величин в десятичные числа с плавающей запятой	EVAL	s, d	Преобразует строковую величину, хранящуюся в s, в десятичное число с плавающей запятой (данные типа REAL). Результат сохраняется в d.		3		7.11.12
	EVALP						
Преобразование 16-битных двоичных данных в ASCII-код (Q)	ASC	s, n, d	Преобразует 16-битные двоичные данные, хранящиеся начиная с s, в шестнадцатеричные коды ASCII и сохраняет результат в d с учетом указанного в n количества знаков.		4		7.11.13
	ASCP						
Преобразование буквенно-числовых строковых величин в ASCII-код (A)	ASC	d	Преобразует указанную буквенно-числовую строковую величину в ASCII-кодировку и сохраняет результат в d.			* 13	7.11.14
Преобразование шестнадцатеричных кодов ASCII в двоичные значения	HEX	s, n, d	Преобразует хранящиеся в s шестнадцатеричные знаки в кодировке ASCII в двоичные значения. Количество конвертируемых знаков устанавливается в n. Результат конвертации сохраняется в d.		4		7.11.15
	HEXP						
Фрагмент строковых данных (правая часть строковой величины)	RIGHT	s, n, d	Считывает n знаков с правой стороны строковой величины (конец строковой величины), хранящейся в s, и сохраняет эти знаки в d.		4		7.11.16
	RIGHTP						
Фрагмент строковых данных (левая часть строковой величины)	LEFT	s, n, d	Считывает n знаков с левой стороны строковой величины (начало строковой величины), хранящейся в s, и сохраняет эти знаки в d.				
	LEFTP						

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Сохранение и перемещение частей строковой величины	MIDR	s1, s2, d	Считывает фрагмент определенной длины строковой величины, хранящейся в s1, и сохраняет его в d. В s2 указана первая позиция считываемого фрагмента.		4		7.11.17
	MIDRP						
	MIDW	s1, s2, d	Считывает фрагмент определенной длины строковой величины, хранящейся в s1, и сохраняет его в d. В s2 указана первая позиция в области d, в которую записывается фрагмент.				
	MIDWP						
Поиск строковых величин	INSTR	s1, s2, n, d	Ищет в строковой величине, указанной в s2, строковую величину, указанную в s1. Поиск начинается с позиции, указанной в n.		5		7.11.18
	INSTRP						
Преобразование числа с плавающей запятой в двоично-десятичный формат	EMOD	s1, s2, d1	Пересчитывает в двоично-десятичный формат число с плавающей запятой (данные типа REAL), хранящееся в s1, с учетом смещения запятой вправо, указанного в s2. Результат сохраняется в d1.		4		7.11.19
	EMODP						
Преобразование числа с плавающей запятой в десятичный формат	EREXP	s1, s2, d1	Пересчитывает двоично-десятичное число с плавающей запятой, хранящееся в s1, в десятичное число с плавающей запятой с учетом указанной в s2 дробной части. Результат сохраняется в d1.		4		7.11.20
	EREXPP						

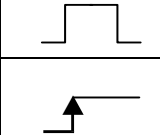
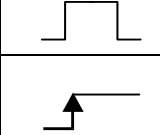
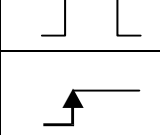
\*: Количество шагов программы зависит от используемых операндов.

Точное количество указано в разделах, посвященных отдельным командам.

2.5.12 Команды особых функций

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Вычисление синуса	SIN	s, d	$SIN(s + 1, s) \rightarrow (d + 1, d)$		3		7.12.1
	SINP						
Вычисление косинуса	COS	s, d	$COS(s + 1, s) \rightarrow (d + 1, d)$		3		7.12.2
	COSP						
Вычисление тангенса	TAN	s, d	$TAN(s + 1, s) \rightarrow (d + 1, d)$		3		7.12.3
	TANP						
Вычисление арксинуса	ASIN	s, d	$ARCSIN(s + 1, s) \rightarrow (d + 1, d)$		3		7.12.4
	ASINP						
Вычисление арккосинуса	ACOS	s, d	$ARCCOS(s + 1, s) \rightarrow (d + 1, d)$		3		7.12.5
	ACOSP						
Вычисление арктангенса	ATAN	s, d	$ARCTAN(s + 1, s) \rightarrow (d + 1, d)$		3		7.12.6
	ATANP						
Пересчет из градусов в радианы	RAD	s, d	$(s + 1, s) \rightarrow (d + 1, d)$ пересчет из градусов в радианы		3		7.12.7
	RADP						
Пересчет из радиан в градусы	DEG	s, d	$(s + 1, s) \rightarrow (d + 1, d)$ пересчет из радиан в градусы		3		7.12.8
	DEGP						
Вычисление квадратного корня	SQR	s, d	$\sqrt{(s + 1, s)} \rightarrow (d + 1, d)$		3		7.12.9
	SQRP						

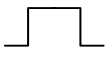

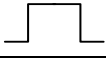
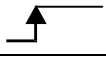

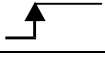
Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка			
					Q	A				
Вычисление экспоненты (степени с основанием e) числа с плавающей запятой	EXP	s, d	$e^{(s+1, s)} \rightarrow (d+1, d)$		3		7.12.10			
	EXPP									
Вычисление натурального логарифма	LOG	s, d	$\text{LOG } e(s+1, s) \rightarrow (d+1, d)$		3		7.12.11			
	LOGP									
Генерирование случайных чисел	RND	d	Сохранение генерированного случайного числа в d.		2		7.12.12			
	RNDP									
Обновление серий случайных чисел	SRND	s	Обновление случайного числа в s.		2					
	SRNDP									
Вычисление квадратного корня 4-значного двоично-десятичного числа	BSQR	s, d	$\sqrt{(s)} \rightarrow (D)+0$ +1 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>целое число</td></tr><tr><td>дробная часть</td></tr></table>	целое число	дробная часть		3		7.12.13	
	целое число									
дробная часть										
BSQRP										
Вычисление квадратного корня 8-значного двоично-десятичного числа	BDSQR	s, d	$\sqrt{(s+1, s)} \rightarrow (D)+0$ +1 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>целое число</td></tr><tr><td>дробная часть</td></tr></table>	целое число	дробная часть		3			
	целое число									
дробная часть										
BDSQRP										
Вычисление синуса двоично-десятичных данных	BSIN	s, d	$\text{Sin } (S) \rightarrow (D)+0$ +1 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>арифм. знак</td></tr><tr><td>целое число</td></tr><tr><td>дробная часть</td></tr></table>	арифм. знак	целое число	дробная часть		3		7.12.14
	арифм. знак									
целое число										
дробная часть										
BSINP										
Вычисление косинуса двоично-десятичных данных	BCOS	s, d	$\text{Cos } (S) \rightarrow (D)+0$ +1 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>Арифм. знак</td></tr><tr><td>целое число</td></tr><tr><td>дробная часть</td></tr></table>	Арифм. знак	целое число	дробная часть		3		7.12.15
	Арифм. знак									
целое число										
дробная часть										
BCOSP										
Вычисление тангенса двоично-десятичных данных	BTAN	s, d	$\text{Tan } (S) \rightarrow (D)+0$ +1 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>ариф. знак</td></tr><tr><td>целое число</td></tr><tr><td>дробная часть</td></tr></table>	ариф. знак	целое число	дробная часть		3		7.12.16
	ариф. знак									
целое число										
дробная часть										
BTANP										

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Вычисление арксинуса двоично-десятичных данных	BASIN	s, d	$ASin(S) \rightarrow$ (D) +0 арифм. знак +1 целое число +2 дробная часть		3		7.12.17
	BASINP						
Вычисление арккосинуса двоично-десятичных данных	BACOS	s, d	$ACos(S) \rightarrow$ (D) +0 арифм. знак +1 целое число +2 дробная часть		3		7.12.18
	BACOSP						
Вычисление арктангенса двоично-десятичных данных	BATAN	s, d	$\lambda Tan(S) \rightarrow$ (D) +0 арифм. знак +1 целое число +2 дробная часть		3		7.12.19
	BATANP						

## 2.5.13 Команды контроля данных

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Ограничение диапазона выходных значений 16/32-битных двоичных данных	LIMIT	s1, s2, s3, d	Если $(s3) < (s1)$ , значение s1 сохраняется в d.		5		7.13.1
	LIMITP		Если $(s1) \leq (s3) \leq (s2)$ , значение s3 сохраняется в d. Если $(s2) < (s3)$ , значение s2 сохраняется в d.				
	DLIMIT	s1, s2, s3, d	Если $((s3) + 1, s3) < ((s1) + 1, s1)$ , значение $((s1) + 1, s1)$ сохраняется в $(d + 1, d)$ .				
	DLIMITP		Если $((s1) + 1, s1) \leq ((s3) + 1, s3) < ((s2) + 1, s2)$ , значение $((s3) + 1, s3)$ сохраняется в $(d + 1, d)$ . Если $((s2) + 1, s2) < ((s3) + 1, s3) < ((s2) + 1, s2)$ , значение $((s2) + 1, s2)$ сохраняется в $(d + 1, d)$ .				
Смещение входного значения 16/32-битных двоичных данных	BAND	s1, s2, s3, d	Если $(s1) \leq (s3) \leq (s2)$ , $0 \rightarrow (d)$		5		7.13.2
	BANDP		Если $(s3) < (s1)$ , $(s3) \rightarrow (s1) \rightarrow (d)$ Если $(s2) < (s3)$ , $(s3) \rightarrow (s2) \rightarrow (d)$				
	DBAND	s1, s2, s3, d	Если $((s1) + 1, s1) \leq ((s3) + 1, s3) \leq ((s2) + 1, s2)$ , $0 \rightarrow (d + 1, d)$				
	DBANDP		Если $((s3) + 1, s3) < (s1 + 1, s1)$ , $((s3) + 1, s3) - ((s1) + 1, s1), \rightarrow (d + 1, d)$ Если $((s2) + 1, s2) < ((s3) + 1, s3)$ , $((s3) + 1, s3) - ((s2) + 1, s2), \rightarrow (d + 1, d)$				
Смещение выходного значения 16/32-битных двоичных данных	ZONE	s1, s2, s3, d	Если $s3 = 0$ : $0 \rightarrow (d)$ Если $s3 > 0$ : $s3 + s2 \rightarrow (d)$		5		7.13.3
	ZONEP		Если $s3 < 0$ : $s3 \rightarrow s1 \rightarrow (d)$				
	DZONE	s1, s2, s3, d	Если $((s3) + 1, s3) = 0$ , $0 \rightarrow (d + 1, d)$ Если $((s3) + 1, s3) > 0$ , $((s3) + 1, s3) + ((s2) + 1, s2), \rightarrow (d + 1, d)$				
	DZONEP		Если $((s3) + 1, s3) < 0$ , $((s3) + 1, s3) + ((s1) + 1, s1), \rightarrow (d + 1, d)$				

### 2.5.14 Команды для переключения блоков регистров файлов

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Переключение между блоками регистра файлов	RSET	s	Переключает с блока регистра файлов, используемого в программе, на блок регистра файлов, адрес которого указан в s.		2		7.14.1
	RSETP						
Переключение между файлами регистров файлов	QDRSET	s	Переключает с используемого в программе файла регистра файлов на файл регистра файлов, указанный в s.		* 2 + n		7.14.2
	QDRSETP						
Переключение между файлами комментариев	QCDSSET	s	Переключает с используемого в программе файла комментариев на файл комментариев, указанный в s.		* 2 + n		7.14.3
	QCDSSETP						

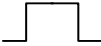
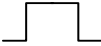
\*:  $n = (\text{Количество знаков в названии файла}/2) = \text{количество дополнительных шагов}$ .  
(Дробная часть округляется вверх).



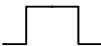

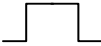
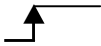

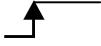


### 2.5.15 Команды для часов

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка																		
					Q	A																			
Считывание данных времени	DATERD	d	часы процессора QnA → <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>d+0</td><td>год</td></tr> <tr><td>d+1</td><td>месяц</td></tr> <tr><td>d+2</td><td>день</td></tr> <tr><td>d+3</td><td>час</td></tr> <tr><td>d+4</td><td>минута</td></tr> <tr><td>d+5</td><td>секунда</td></tr> <tr><td>d+6</td><td>день недели</td></tr> </table>	d+0	год	d+1	месяц	d+2	день	d+3	час	d+4	минута	d+5	секунда	d+6	день недели		2		7.15.1				
	d+0			год																					
d+1	месяц																								
d+2	день																								
d+3	час																								
d+4	минута																								
d+5	секунда																								
d+6	день недели																								
DATERDP																									
Запись данных времени	DATEWR	s	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>d+0</td><td>год</td></tr> <tr><td>d+1</td><td>месяц</td></tr> <tr><td>d+2</td><td>день</td></tr> <tr><td>d+3</td><td>час</td></tr> <tr><td>d+4</td><td>минута</td></tr> <tr><td>d+5</td><td>секунда</td></tr> <tr><td>d+6</td><td>день недели</td></tr> </table> → часы процессора QnA	d+0	год	d+1	месяц	d+2	день	d+3	час	d+4	минута	d+5	секунда	d+6	день недели		2		7.15.2				
	d+0			год																					
d+1	месяц																								
d+2	день																								
d+3	час																								
d+4	минута																								
d+5	секунда																								
d+6	день недели																								
DATEWRP																									
Сложение данных времени	DATE +	s1, s2, d	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>s1</td><td>час</td></tr> <tr><td></td><td>минута</td></tr> <tr><td></td><td>секунда</td></tr> </table> + <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>s2</td><td>час</td></tr> <tr><td></td><td>минута</td></tr> <tr><td></td><td>секунда</td></tr> </table> → <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>d</td><td>час</td></tr> <tr><td></td><td>минута</td></tr> <tr><td></td><td>секунда</td></tr> </table>	s1	час		минута		секунда	s2	час		минута		секунда	d	час		минута		секунда		4		7.15.3
	s1			час																					
	минута																								
	секунда																								
s2	час																								
	минута																								
	секунда																								
d	час																								
	минута																								
	секунда																								
DATE + P																									
Вычитание данных времени	DATE -	s1, s2, d	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>s1</td><td>час</td></tr> <tr><td></td><td>минута</td></tr> <tr><td></td><td>секунда</td></tr> </table> - <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>s2</td><td>час</td></tr> <tr><td></td><td>минута</td></tr> <tr><td></td><td>секунда</td></tr> </table> → <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>d</td><td>час</td></tr> <tr><td></td><td>минута</td></tr> <tr><td></td><td>секунда</td></tr> </table>	s1	час		минута		секунда	s2	час		минута		секунда	d	час		минута		секунда		4		7.15.4
	s1			час																					
	минута																								
	секунда																								
s2	час																								
	минута																								
	секунда																								
d	час																								
	минута																								
	секунда																								
DATE - P																									
Изменение формата данных времени с "ч, мин, сек" на секунды	SECOND	s, d	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>s</td><td>час</td></tr> <tr><td></td><td>минута</td></tr> <tr><td></td><td>секунда</td></tr> </table> → <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>d</td><td>секунда</td></tr> </table>	s	час		минута		секунда	d	секунда		3		7.15.5										
	s			час																					
	минута																								
	секунда																								
d	секунда																								
SECONDP																									
Изменение формата данных времени с секунд на "ч, мин, сек"	HOUR	s, d	<table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>s</td><td>секунда</td></tr> </table> → <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>d</td><td>час</td></tr> <tr><td></td><td>минута</td></tr> <tr><td></td><td>секунда</td></tr> </table>	s	секунда	d	час		минута		секунда														
	s			секунда																					
d	час																								
	минута																								
	секунда																								
HOURP																									

### 2.5.16 Команды для периферийной аппаратуры

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Вывод сообщений на периферийные устройства	MSG	s	Выводит строковую величину, находящуюся в s, в виде сообщения на периферийное устройство, указанное в режиме терминала.		2		7.16.1
Ввод данных с клавиатуры периферийного устройства	PKEY	d	Введенные с клавиатуры данные (знаки) считываются из периферийного устройства, определенного в режиме терминала, и записываются в d в кодировке ASCII.		2		7.16.2


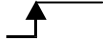


### 2.5.17 Программные команды

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Переключение программ в резервный режим	PSTOP	s	Переключает программный файл, указанный в s, в резервный режим.		* 2 + n		7.17.1
	PSTOPP						
Переключение программ в резервный режим со сбросом выходов	POFF	s	Переключает программный файл, указанный в s, в резервный режим со сбросом выходов.		* 2 + n		7.17.2
	POFFP						
Переключение программ в режим одного прогона программы	PSCAN	s	Переключает программный файл, указанный в s, в режим одного прогона программы.		* 2 + n		7.17.3
	PSCANP						
Переключение программ в режим замедленного выполнения	PLOW	s	Переключает программный файл, указанный в s, в режим низкой скорости выполнения.		* 2 + n		7.17.4
	PLOWP						

\*: n = (Количество знаков в названии программы/2) = количество дополнительных шагов (Дробная часть округляется вверх).

## 2.5.18 Прочие команды

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Сброс сторожевого таймера	WDT		Сбрасывает сторожевой таймер (WDT) в основной программе.		1	1	7.18.1
	WDTP						
Установка и сброс флага переноса	STC		Во флаг переноса записывается цифра (0 или 1), переносимая при процессах вращения и сдвига битов.			1	7.18.2
	CLC		Эта команда сбрасывает флаг переноса.				
Задание циклов выполнения	DUTY	n1, n2, d			4	7	7.18.3
Непосредственное считывание одного байта	ZRRDB	n, d			3		7.18.4
	ZRRDBP						
Непосредственная запись одного байта	ZRWRB	n, s			3		7.18.5
	ZRWRBP						
Сохранение косвенного адреса	ADRSET	s, d	Сохраняет косвенный адрес операнда, указанного в s, в операнде, указанном в d.		3		7.18.6
	ADRSETP						
Ввод числовых значений с помощью клавиш	KEY	s, n, d1, d2	Эта команда поддерживает ввод с помощью клавиш 8 знаков в кодировке ASCII на входы (X), указанные в s. Введенные значения кодируются в шестнадцатеричном виде и сохраняются в d1.		5		7.18.7
Сохранение содержимого индексных регистров в регистре	ZPUSH	d	Сохраняет содержимое индексных регистров с Z0 по Z15 в d.		2		7.18.8
	ZPUSHP						

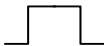
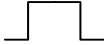

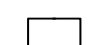
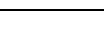


Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Восстановление содержимого индексных регистров из регистра	ZPOP	d	Восстанавливается содержимое индексных регистров с Z0 по Z15, сохраненное в d.		2		7.18.9
	ZPOPP						
Запись данных в регистр EEPROM	EROMWR	s, n, d1,d2	Записывает хранящиеся в s слова данных, количество которых указано в n, в регистр файлов EEPROM, указанный в d1.		6		7.18.10
	EROMWRP						

## 2.6 Команды сетевой передачи данных



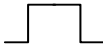
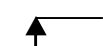
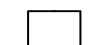

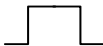
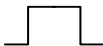
### 2.6.1 Команды обновления сетевых данных

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Обновление сетевых данных	ZCOM	Jn	Обновление данных в модулях сетевой коммуникации		5		8.5.1
		Un					

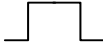

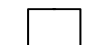
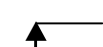
### 2.6.2 Специализированные команды передачи данных (совместимые с серией QnA)

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Сетевая передача данных (совместимая с серией QnA)	READ	Jn, s1, s2, d1, d2	Считывание данных центрального процессора из целевых станций		9		8.6.1
		Un, s1, s2, d1, d2					
	SREAD	Jn, s1, s2, d1, d2, d3	Запись данных центрального процессора в целевые станции		10		8.6.2
		Un, s1, s2, d1, d2, d3					
	WRITE	Jn, s1, s2, d1, d2	Запись данных центрального процессора в целевые станции		10		8.6.3
		Un, s1, s2, d1, d2					
	SWRITE	Jn, s1, s2, d1, d2, d3	Запись данных центрального процессора в целевые станции		11		8.6.4
		Un, s1, s2, d1, d2, d3					
	SEND	Jn, s1, s2, d	Передача данных в модули сетевой коммуникации		8		8.6.5
		Un, s1, s2, d					
	RECV	Jn, s, d1, d2	Считывание данных, переданных с помощью команды SEND		8		8.6.6
		Un, s, d1, d2					
	REQ	Jn, s1, s2, d1, d2	Запрос данных у других станций		8		8.6.7
		Un, s1, s2, d1, d2					
ZNFR	Jn, s1, s2, d	Считывание данных из специальных модулей удаленных станций		8		8.6.8	
	Un, s1, s2, d						
ZNT0	Jn, s1, s2, d	Запись данных в специальные модули удаленных станций		8		8.6.9	
	Un, s1, s2, d						

### 2.6.3 Команды передачи данных (совместимые с серией "A")

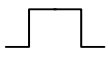

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Сетевая передача данных (совместимая с серией "A")	J.ZNRD	Jn, n1, s, n2, d1, d2	Считывание данных процессоров QnA из целевых станций		32		8.7.1
	JP.ZNRD		Считывание данных из локальных станций				
	J.ZNWR	Jn, n1, s, n2, d1, d2	Запись данных из процессора QnA в целевую станцию		32		8.7.2
	JP.ZNWR		Запись данных в локальную станцию				
	LRDP	s, n1, n2, d	Считывание данных из локальной станции (только серия "A")			11	8.7.3
	LWTP	Jn, s, d1, d2	Запись данных в локальную станцию (только серия "A")			11	8.7.4
	RFRP	n1, n2, n3, d	Считывание данных из специального модуля удаленной станции		11	11	8.7.5
	G.RFRP	Un, n1, n2, d1, d2					
	RTOP	s, n1, n2, n3	Запись данных в специальные модули удаленной станции		11	11	8.7.6
G.RTOP	Un, n1, s, n2, d1						

### 2.6.4 Информация маршрутизации

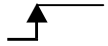

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Считывание и запись информации маршрутизации	Z.RTREAD	n, d	Считывание информации маршрутизации сети n и сохранение данных в d		7		8.8.1
	ZP.RTREAD						
	Z.RTWRITE	s, n	Запись информации маршрутизации из s в сеть n		8		8.8.2
	ZP.RTWRITE						

## 2.7 Команды для центральных процессоров MELSEC серии "System Q"



### 2.7.1 Считывание информации модуля

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Считывание информации из модуля	UNIRD	n1, d, n2	Информация модуля считывается, начиная с адреса ввода-вывода, указанного в n1, и записывается, начиная с адреса, указанного в d. Длина данных указывается в n2.		4		9.1.1
	UNIRDP						

### 2.7.2 Отладка и поиск ошибок

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Установка и сброс контроля (трассировки)	TRACE		Если установлены маркеры SM800, SM801 и SM802, эта команда выполняет промежуточное сохранение данных операндов, выбранных с помощью программатора, на карте памяти.		1		9.2.1
	TRACER		Стирает данные, сохраненные с помощью команды TRACE.		1		9.2.1

### 2.7.3 Запись в файлы и считывание из файлов

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Запись данных в файл	SP.FWRITE	u0, s0, d0, s1, s2, d1	Записывает данные в указанный файл		11		9.3.1
Считывание данных из файла	SP.FREAD	u0, s0, d0, s1, d1, d2	Считывает данные из указанного файла		11		9.3.2

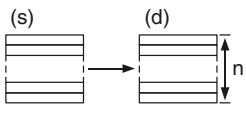
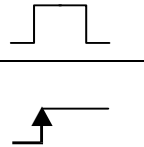
### 2.7.4 Программные команды

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Загрузка программы с носителя данных	PLOADP	s, d	Загружает программный файл с носителя данных (не являющегося дисководом 0) в дисковод 0 и переключает программу в резервный режим.		3		9.4.1
Стирание программы, находящейся в резервном режиме	PUNLOADP	s, d	Стирает файл программы, находящейся в резервном режиме на дисководе 0.		3		9.4.2
Стирание программы, находящейся в резервном режиме, и загрузка программы с носителя данных	PSWAPP	s1, s2, d	Стирает файл s1 программы, находящийся на дисководе 0 в резервном режиме. Затем с носителя данных (не являющегося дисководом 0) в дисковод 0 передается программный файл s2 и этой файл переключается в резервный режим.		4		9.4.3

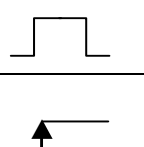
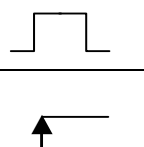
Эти функции имеются только в среде GX Developer, так как GX IEC Developer не поддерживает файловую концепцию.



## 2.7.5 Команды передачи

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Передача блоков данных с высокой скоростью	RBMOV	s, d, n			4		9.5.1
	RBMOPV						

## 2.7.6 Команды для обмена данными в многопроцессорном режиме

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Запись данных в общую область памяти	S.TO	s1, s2, s3, s4, d	Данные записываются в общую область памяти того центрального процессора, который выполняет команду S.TO.		5		9.6.1
	S.TOP				5		
Считывание данных из общей области памяти другого центрального процессора.	FROM	n1, n2, n3, d	Данные из общей области памяти другого ЦП передаются в центральный процессор, в котором выполняется команда FROM.		5		9.6.2
	FROMP				5		
Обновление общей области памяти	COM		Обновляет общую область памяти для многопроцессорного режима.		1		6.7.3

## 2.8 Специальные команды для процессора Q4AR

### 2.8.1 Команды установки режима







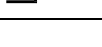
Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Установка режима запуска центрального процессора	SGMODE	s1, s2	Выбор между новым запуском и перезапуском (горячим запуском)				10.1.1
Режим при переключении	CGMODE	s	Выбор режима при переключении с активного ЦП на резервный ЦП				10.1.2

### 2.8.2 Команды передачи

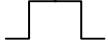



Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Передача данных в резервный ЦП	TRUCK	s	Передача данных операндов из активного ЦП в резервный ЦП				10.2.1
Обмен данными со специальными модулями	SPREF	s	Поблочное считывание данных из буферной памяти специальных модулей и запись в эту память				10.2.2

## 2.9 Команды для специальных модулей









### 2.9.1 Команды для модулей последовательных интерфейсов

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Передача данных из модуля интерфейса в центральный процессор контроллера	BUFRCVS	„Un“, n1, d1	Принятые данные передаются из модуля интерфейса в центральный процессор ПЛК в рамках программы прерывания.				11.1.1
Считывание кадра данных, определенного пользователем	GETE	Un, s1, s2, d	Считывает из модуля интерфейса кадр данных, определенный пользователем				11.1.2
	GETEP						
Запись или стирание кадра данных, определенного пользователем	PUTE	Un, s1, s2, d	Запись определенного пользователем кадра данных в модуль интерфейса или его стирание в модуле интерфейса				11.1.3
	PUTEP						
Передача данных	PRR	Un, s, d	Передача данных через модуль интерфейса с использованием кадра данных, определенного пользователем				11.1.4
	PRRP						

2.9.2 Команды для модулей Profibus/DP

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Считывание данных	BBLKRD	„Un“, n1, n2, d	Данные считываются из буферной памяти модуля Profibus/DP и записываются в модуль центрального процессора.				11.2.1
	BBLKRD						
Запись данных	BBLKWR	„Un“, n1, n2, s	Данные из центрального процессора записываются в буферную память модуля Profibus/DP.				11.2.2
	BBLKWR						

### 2.9.3 Команды для модулей Ethernet

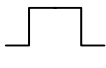
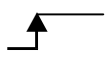
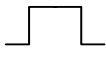
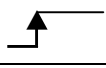
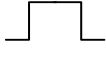
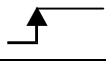
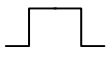
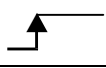
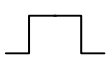
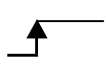
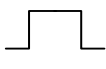
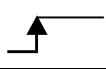
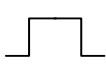
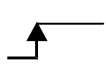
Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Считывание данных из фиксированных буферов	BUFRCV	„Un“, s1, s2, d1, d2	При коммуникации с использованием фиксированных буферов принятые данные считываются из модуля ETHERNET.				11.3.1
	BUFRCVS	„Un“, s1, d1					11.3.2
Запись данных в фиксированные буферы	BUFSND	„Un“, s1, s2, s3, d1	Данные из центрального процессора передаются в модуль Ethernet				11.3.3
Открытие канала связи	OPEN	„Un“, s1, s2, d1	Установление связи				11.3.4
Закрытие канала связи	CLOSE	„Un“, s1, s2, d1	Прекращение связи				11.3.5
Стирание ошибок	ERRCLR	„Un“, s1, d1	Стирание кодов ошибок в буферной памяти и выключение светодиода "ERR." модуля Ethernet				11.3.6
Считывание кода ошибки	ERRRD	„Un“, s1, d1	Считывание кодов ошибок из буферной памяти				11.3.7
Повторная инициализация модуля Ethernet	UINI	„Un“, s1, d1	Указанный с помощью "Un" модуль Ethernet инициализируется еще раз.				11.3.8



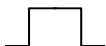

### 2.9.4 Команда для MELSECNET/10

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Установление пар станций	PAIRSET	Jn, s1	Установление станций, соединяемых при дуплексном режиме				11.4.1

2.9.5 Команды для CC-Link

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Передача параметров сети (серия "A")	RLPA	n, d1, d2	Передача параметров сети в мастер-станцию CC-Link		—	23	11.5.1
	RLPA_P						
Передача параметров сети (System Q)	RLPASET	Un, s1 до s5, d1				—	11.5.2
	RLPASET_P						
Передача параметров автоматич. обновления (серия "A")	RRPA	n, d	Передача параметров автоматического обновления операндов		—	20	11.5.3
	RRPA_P						
Считывание данных из буферной памяти или другого центрального процессора контроллера (серия "A")	RIRD	n1, n2, d1, d2	Считывание данных из буферной памяти модуля CC-Link другой станции или из центрального процессора контроллера этой станции		—	26	11.5.4
	RIRD_P						
Считывание данных из буферной памяти или другого центрального процессора контроллера (серии QnA и "System Q")	RIRD	Un, s, d1, d2			8	—	11.5.5
	RIRD_P						
Запись данных в буферную память или другой центральный процессор контроллера (серия "A")	RIWT	n1, n2, d1, d2	Запись данных в буферную память модуля CC-Link другой станции или в центральный процессор ПЛК этой станции		—	26	11.5.6
	RIWT_P						

Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка
					Q	A	
Запись данных в буферную память или другой центральный процессор контроллера (серия QnA, "System Q")	RIWT	Un, s, d1, d2	Запись данных в буферную память модуля CC-Link другой станции или в центральный процессор ПЛК этой станции		8	—	11.5.7
	RIWT_P						
Считывание данных из интеллектуальной станции (серия "A")	RICV	n1, n2, d1, d2, d3	Данные считываются из буферной памяти интеллектуальной станции CC-Link с квитированием связи.		—	29	11.5.8
	RICV_P						
Считывание данных из интеллектуальной станции (серия QnA, "System Q")	RICV	Un, s1, s2, d1, d2	Данные считываются из буферной памяти интеллектуальной станции CC-Link с квитированием связи.		10	—	11.5.9
	RICV_P						
Запись данных в интеллектуальную станцию (серия "A")	RISEND	n1, n2, d1, d2, d3	Данные записываются в буферную память интеллектуальной станции CC-Link с квитированием связи.		—	29	11.5.10
	RISEND_P						
Запись данных в интеллектуальную станцию (серия QnA, "System Q")	RISEND	Un, s1, s2, d1, d2	Данные записываются в буферную память интеллектуальной станции CC-Link с квитированием связи.		10	—	11.5.11
	RISEND_P						
Запись данных в автоматически обновляемую память (серия "A")	RITO	n1, n2, n3, n4, d1	Запись данных из центрального процессора контроллера в автоматически обновляемую область буферной памяти главной станции CC-Link. Затем эти данные передаются в указанную станцию.		—	29	11.5.12
	RITO_P						
Запись данных в автоматически обновляемую память (серия QnA, "System Q")	RITO	Un, n1, n2, n3, d	Запись данных из центрального процессора контроллера в автоматически обновляемую область буферной памяти главной станции CC-Link. Затем эти данные передаются в указанную станцию.		9	—	11.5.13
	RITO_P						

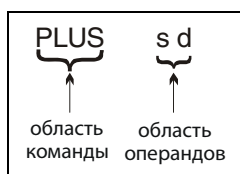
Группа	Команда	Переменные	Значение	Выполнение	Шагов		Ссылка	
					Q	A		
Считывание данных из автоматически обновляемой памяти (серия "A")	RIFR	n1, n2, n3, n4, d1	Считывание данных, записанных другой станцией в автоматически обновляемую область буферной памяти главной станции CC-Link.		—	29	11.5.14	
	RIFR_P							
Считывание данных из автоматически обновляемой памяти (серия QnA, "System Q")	RIFR	Un, n1, n2, n3, d			9	—		11.5.15
	RIFR_P							



## 3 Конфигурация команд

### 3.1 Структура команды

Большинство команд состоят из названия команды и указания операндов. Некоторые команды не нуждаются ни в одном операнде и поэтому состоят только из названия команды.



#### Название команды

Название команды описывает функцию команды.

PLUS  $\hat{=}$  сложение

#### Операнды

Здесь указываются используемые константы или переменные. Указанные операнды могут состоять из трех частей: источника данных (s), места назначения (d) и количества (n).

#### 3.1.1 Источник данных (s)

- Источник данных обозначает операнды, обрабатываемые этой командой. В 16-битных командах источник данных обозначается буквой s. В 32-битных командах он обозначается s + 1 и s.
- В качестве источника данных можно указывать константы или переменные.

#### Константа

Константы обозначают постоянное числовое значение, обрабатываемое командой. Это значение установлено программой, написанной пользователем, на постоянной основе и во время выполнения программы не может быть изменено. Каждую переменную, используемую для хранения константы, рекомендуется индексировать.

#### Переменная

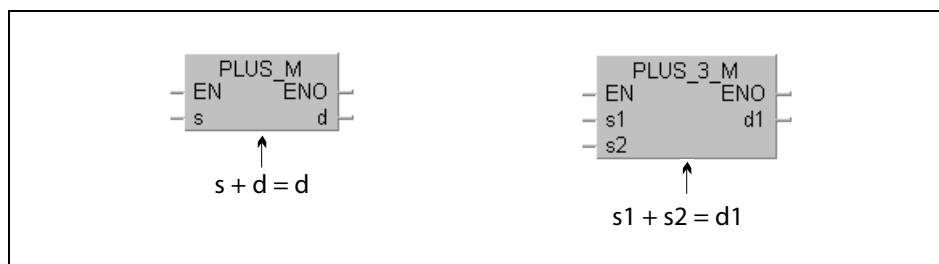
Переменные обозначают операнд, в котором сохранены данные, обрабатываемые командой (см. также раздел 3.4).

Перед выполнением команды данные должны быть сохранены в операнде. Сохраненные данные можно изменять во время выполнения программы.

### 3.1.2 Место назначения (d)

- Место назначения обозначает операнды, в которых данные сохраняются после обработки.  
В 16-битных командах место назначения обозначается буквой d.  
В 32-битных командах место назначения обозначается в виде d + 1 и d.  
Для некоторых команд с 2 операндами необходимо, чтобы обрабатываемое значение сохранялось в месте назначения d перед выполнением команды.  
В этом случае результат обработки сохраняется в том же самом операнде d.

Пример: команда сложения 16-битных двоичных данных



- Для сохранения данных всегда должно быть указано место назначения в виде какого-либо операнда.

### 3.1.3 Количество (n)

- С помощью n указывается, сколько операндов должны использоваться или сколько раз должна быть выполнена команда.

Пример: команда передачи блоков



- Значение в n может находиться в пределах от 0 до 32767. Если количество установлено на 0, команда не выполняется.

## 3.2 Обозначения, принятые в названиях команд

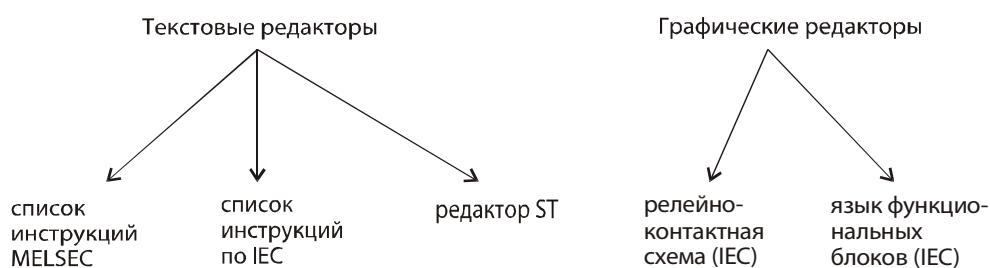
По названию команды можно определить некоторые ее особенности.

### 3.2.1 16/32-битные и импульсные команды

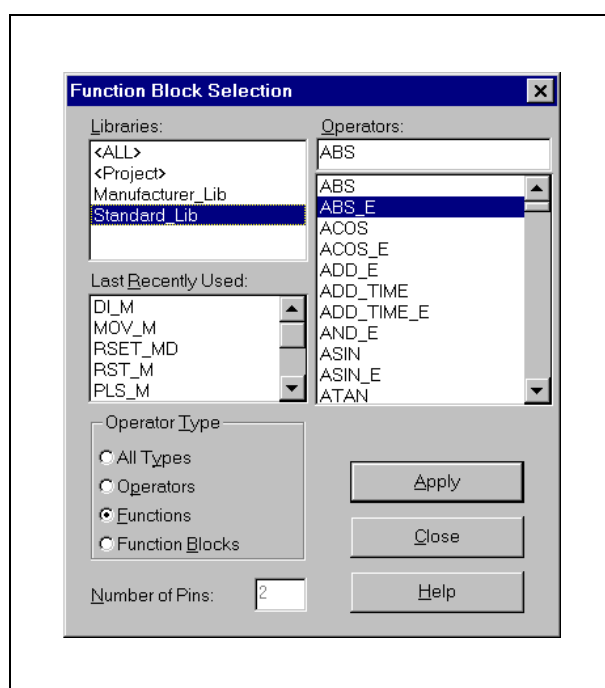
SORT	16-битная обработка
<u>SORT</u>	16-битная обработка по фронту импульса
<u><u>SORT</u></u>	32-битная обработка
<u><u><u>SORT</u></u></u>	32-битная обработка по фронту импульса

### 3.2.2 MELSEC и МЭК (IEC)

В среде программирования GX IEC Developer имеются различные редакторы:

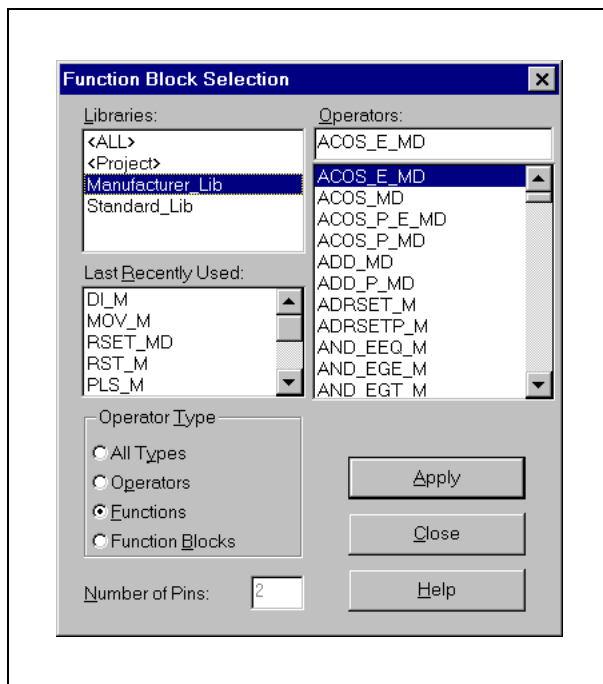


Написание команд в этих редакторах различается.



При выборе команды в среде GX IEC Developer появляется следующее диалоговое окно.

В окне "Библиотеки" можно выбрать, какие команды отображаются для выбора в окне "Операторы":  
 ВСЕ (ALL): команды MELSEC и стандартные команды МЭК (IEC)  
 Проект (Project): собственные функции и функциональные блоки  
 Изготовитель (Manufacturer\_Lib): команды MELSEC  
 Стандарт (Standard\_Lib): команды, соответствующие стандарту Международной электротехнической комиссии (IEC)



Например, если выбрана "Библиотека изготовителя" (Manufacturer\_Lib), то при выборе команды в GX IEC Developer появляется следующее окно. В нем содержится перечень так называемых "адаптированных" команд, т. е. команд, приспособленных к контроллерам MELSEC.

"Оригинальные" и "адаптированные" команды действуют одинаково. Различается только их написание.

**Значение окончаний в редакторе IEC:**

Окончание в редакторе IEC	Значение
_M	команда MELSEC
_P_M	импульсный вариант команды
_MD	специализированная прикладная команда MELSEC (Dedicated Instruction) (см. также раздел 3.3)
_P_MD	импульсный вариант специализированной прикладной команды
_K_MD	специализированная прикладная команда, в которой используется константа
_K_P_MD	импульсный вариант и применение константы в специализированной прикладной команде
_S_MD	специализированная прикладная команда MELSEC для центральных процессоров серии "System Q"
_P_S_MD	импульсный вариант специализированной прикладной команды для центральных процессоров серии "System Q"

### 3.2.3 Прочие особенности написания

В нижеследующей таблице перечислены символы, встречающиеся в некоторых командах в редакторе MELSEC. В правом столбце перечислены соответствующие названия в редакторе IEC.

Пример: Редактор MELSEC      Редактор IEC  
           LD\$>                    LD\_STRING\_GT\_M

Редактор MELSEC	Редактор IEC
\$	STRING
=	EQ
<>	NE
<=	LE
<	LT
>=	GE
>	GT
+	PLUS
-	MINUS
x	MULTI
/	DIVID

### 3.2.4 Написание команд в этом руководстве

В разделах с пятого по восьмой, в которых команды описываются подробно, используются оба редактора, т. е. оба написания. В заголовке указывается "оригинальная" команда MELSEC, т. е. команда в таком виде, в каком она отображается на языке IL для MELSEC.

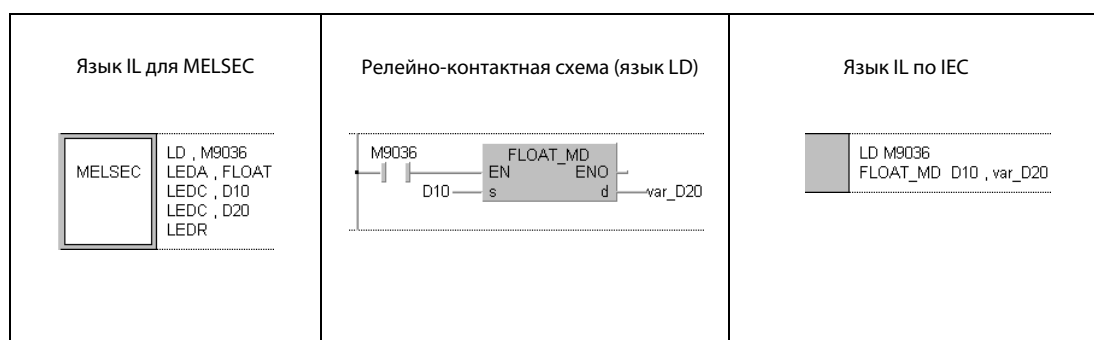
**ПРИМЕЧАНИЕ** *В начале описания каждой группы команд всегда дается обзорная таблица, содержащая оба написания.*

### 3.3 Программирование специализированных команд

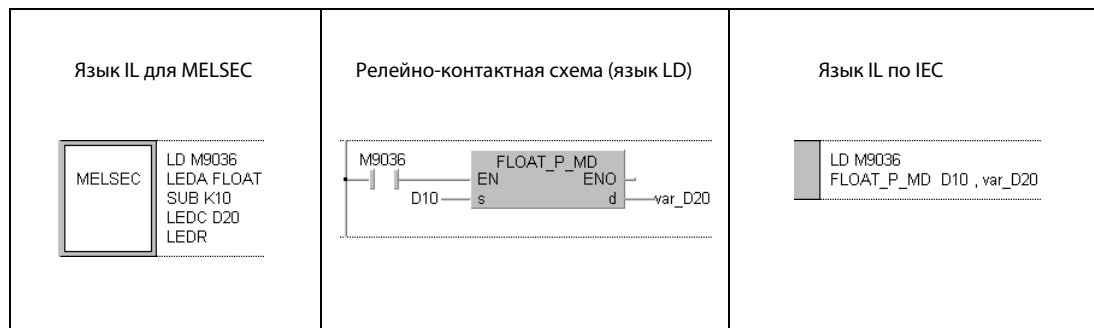
Специализированные команды программирования (Dedicated Instructions) – это "адаптированные" команды, которые не только отличаются от "оригинальных" команд MELSEC по написанию, но и требуют особых методов программирования в зависимости от применяемых центральных процессоров.

Например, чтобы получить функцию команды FLOAT\_MD в редакторе MELSEC для процессоров серии "A", необходим особый порядок действий. В редакторе MELSEC команду FLOAT\_MD необходимо запрограммировать вместе с командами LEDA, LEDC, LEDR. В редакторах IEC специализированные команды можно программировать обычным образом.

Пример: программирование команды FLOAT\_MD (обычный вариант, 16 бит)



Пример: программирование команды FLOAT\_P\_MD (импульсный вариант, 16 бит, применение константы в операнде s)



Более подробную информацию по программированию специализированных прикладных команд вы найдете в следующих руководствах:

- Руководство по GX IEC Developer
- Programming Manual (Dedicated Instructions)

### 3.4 Программирование переменных

В большинстве команд помимо названия команды требуется указывать операнды, содержащие переменные. Эти переменные содержат значения, необходимые для выполнения команды.

В зависимости от редактора, выбранного в GX IEC Developer, переменные программируются по-разному.

#### В редакторе MELSEC:

Регистры данных D100 и D10 можно присваивать значениям непосредственно, обозначая переменные как D100 и D10.

Подключенный контроллер автоматически распознает, что имеются в виду следующие операнды.

D100 = D100 и D101  
 D10 = D10, D11, D12, D13

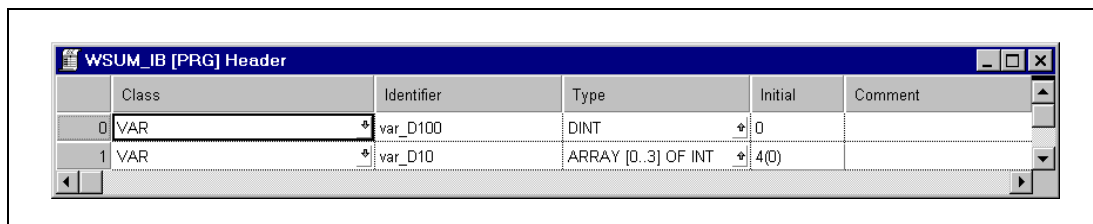
#### В редакторе IEC:

В редакторе IEC непосредственное указание операндов возможно только в том случае, если имеются в виду действительно только эти операнды.

Пример: AND D10

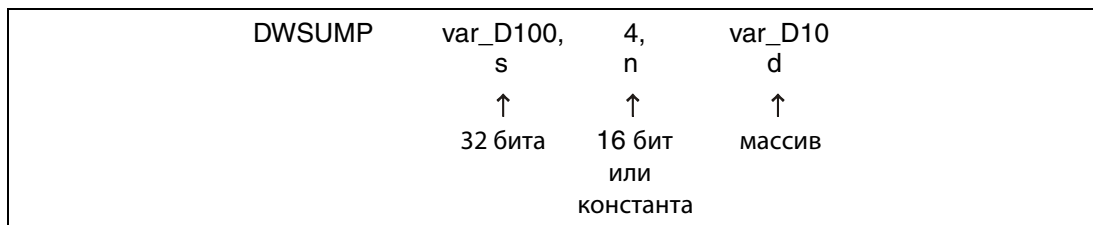
Для применения команды DWSUMP\_M необходимо предварительно объявить переменные в заголовке программного компонента (POU).

Пример: заголовок программы на языке IL по стандарту IEC



Здесь в качестве идентификаторов (имен переменных) указаны "var\_D100" и "var\_D10". Речь не идет непосредственно об операндах D100 и D10. Программируемый контроллер присваивает этим переменным свободные области регистра.

Пример: DWSUMP



Переменная var\_D100 имеет тип DINT (32 бита). Переменная var\_D10 имеет тип ARRAY. Массив содержит четыре 16-битных регистра типа INT (см. также раздел 3.5.2 "Адресация массивов и регистров в GX IEC Developer").

**Написание**

Имена var\_D100 или var\_D10 сразу указывают на то, что это не сами операнды, а идентификаторы. В этих случаях всегда необходимо объявлять переменную! Если программирование команды возможно только с объявлением переменной, в тексте руководства всегда имеется соответствующее примечание.

**ПРИМЕЧАНИЕ** *В качестве идентификатора можно применять любое имя (например, Motor1, Лампа). В примере намеренно выбраны имена var\_D100 или var\_D10 – для более наглядного сравнения с программированием в редакторе MELSEC.*

В начале описания каждой команды дается обзор типов данных (таблица переменных). Ниже приведен пример таблицы переменных для команды DWSUM, раздел 7.5.14).

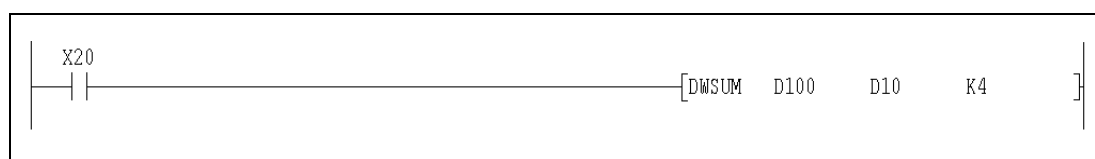
Операнд	Значение команды	Тип данных	
		MELSEC	МЭК (IEC)
s	Первый адрес операнда, в котором сохранены суммируемые данные.	BIN, 32 бита	ANY32
d	Первый адрес операнда, в котором сохраняется результат.	BIN, 64 бита	массив [1..4] данных типа ANY16
n	Количество суммируемых блоков данных.	BIN, 16 бит	ANY16

**В среде GX Developer:**

К регистрам данных D100 и D10 можно обращаться непосредственно, обозначая переменные D100 и D10.

Подключенный программируемый контроллер автоматически распознает, что имеются в виду следующие операнды.

- D100 = D100 и D101
- D10 = D10, D11, D12, D13





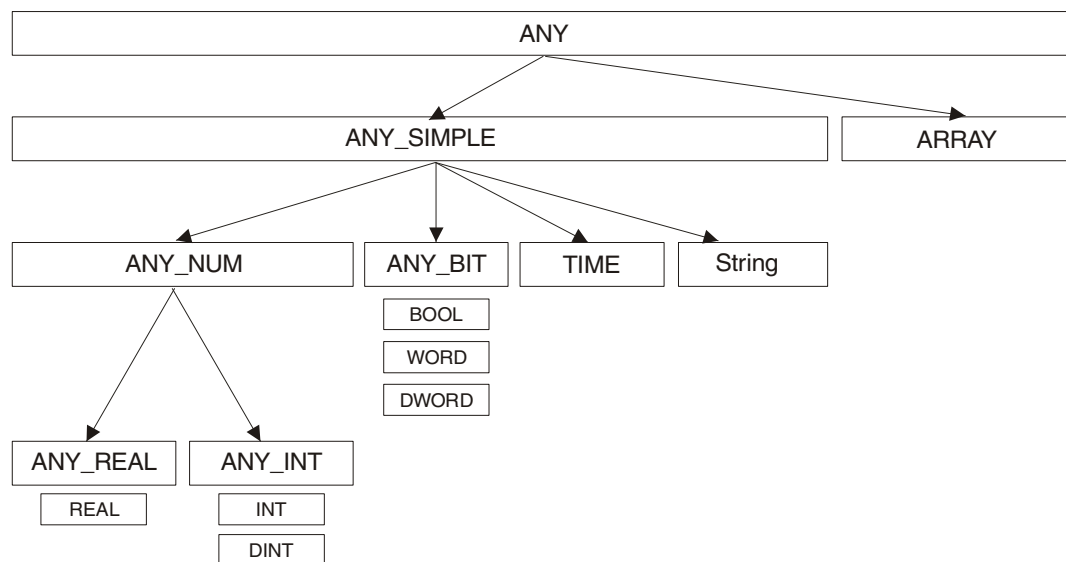
### 3.5 Типы данных

Тип данных задает количество битов и правила работы с ними, а также диапазон значений переменных.

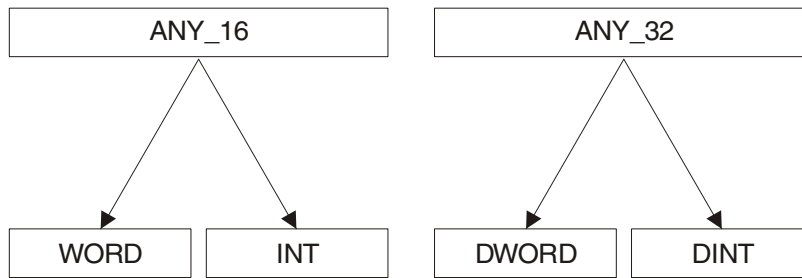
Имеются следующие типы данных:

Тип данных		Диапазон	Размер	Применимый ЦП
BOOL	булевый	0 (FALSE), 1 (TRUE)	1 бит	серия "А" серия "Q" "System Q"
INT	целое число	от -32.768 до 32.767	16 битов	
DINT	целое число двойной длины	от -2.147.483.648 до 2.147.483.647	32 бита	
WORD	цепочка из 16 битов	от 0 до 65.535	16 битов	
DWORD	цепочка из 32 битов	от 0 до 4.294.967.295	32 бита	
REAL	число с плавающей запятой	3.4 +/- 38 (7 разрядов)	32 бита	
TIME	значение времени	от T#-24 д-0 ч 31 м 23 с 648.00 мс до T#24 д 20 ч 31 м 23 с 647.00 мс	32 бита	серия "Q" "System Q"
STRING	строковая величина	макс. 50 знаков		

#### Иерархия типов данных ANY



**Иерархия типов данных ANY16 и ANY32**



Тип данных	Значение
ANY	любой тип данных
ANY_SIMPLE	простой тип данных
ANY_NUM	числовой тип данных
ANY_REAL	число с плавающей запятой
ANY_INT	целочисленный тип данных
ANY_BIT	тип данных для работы с битами
ANY_16	любой 16-битный тип данных
ANY_32	любой 32-битный тип данных
TIME	время
STRING	строковая величина
REAL	число с плавающей запятой
INT	целое число
DINT	целое число двойной длины
BOOL	булева величина
WORD	слово (16 бит)
DWORD	двойное слово (32 бита)
ARRAY	массив

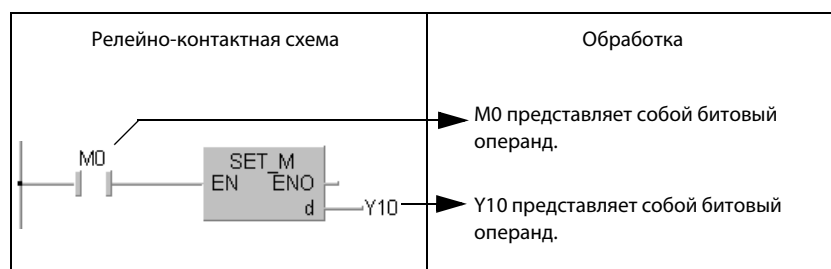
### 3.5.1 Обработка данных

#### Обработка битовых данных

Битовый операнд (X, Y, M, K, S, B или F) может принимать два состояния (ВКЛ = 1 или ВЫКЛ = 0). Таким образом, его состояние можно выразить одним битом (1 или 0). Обработка бита происходит всякий раз, когда программа обращается к определенному битовому операнду. При обработке 16- или 32-битных команд несколько битовых операндов обобщаются в блоки, состоящие из 16 или 32 адресов операндов.

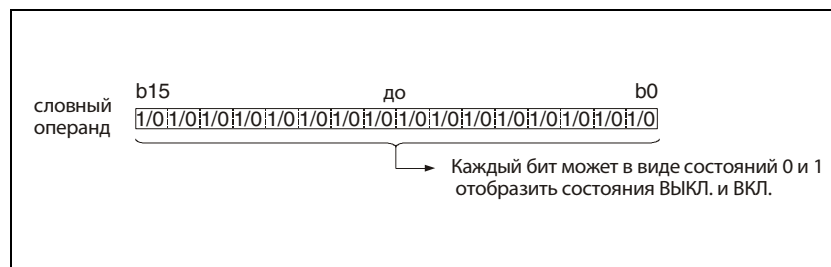
- Применение битовых операндов

Битовый операнд (например, вход, выход, маркер) состоит из одного бита.

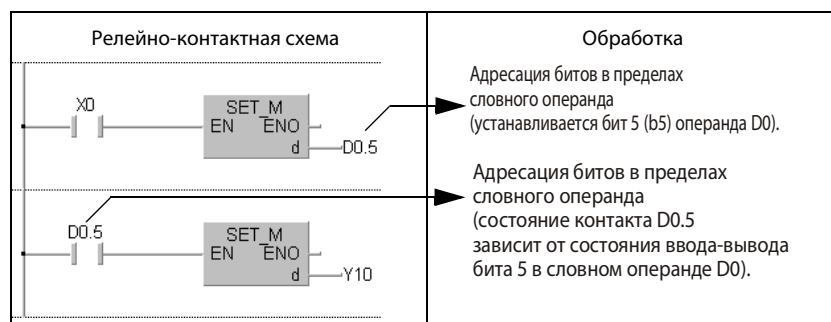


- Применение словных операндов

В центральных процессорах MELSEC серий QnA и "System Q" имеется возможность адресовать каждый отдельный бит в словном операнде.



Битовая адресация осуществляется в шестнадцатеричном формате. Например, если требуется указать бит 5 (b5) операнда D0, адрес имеет вид "D0.5". Бит 10 адресуется в виде "D0.A". Адресация отдельных битов таймеров, счетчиков и фиксируемых таймеров не возможна.



● Применение битовых блоков

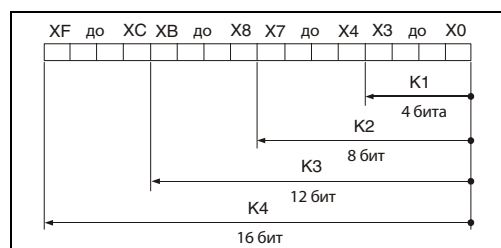
Отдельные биты можно обобщить в блоки по 4 бита и, таким образом, работать со словными данными. Подробное разъяснение имеется на следующих страницах, раздел "Обработка словных данных (16/32 бита)".

**Обработка словных данных (16 бит)**

● Применение битовых операндов

Битовые операнды можно использовать для словных данных. Для этого необходимо установить количество битовых операндов (адресов). Возможна обработка до 16 битов в виде блоков по 4 бита. Длина блока устанавливается с помощью обозначений от K1 до K4.

- K1X0            4 адреса с X0 по X3
- K2X0            8 адресов с X0 по X7
- K3X0            12 адресов с X0 по XB
- K4X0            16 адресов с X0 по XF

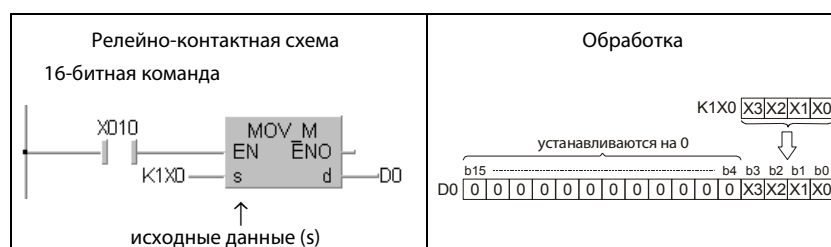


● Указание битовых блоков для s

В следующей таблице указан диапазон исходных данных, если для обозначения исходных данных s задается длина блока.

Длина блока	16-битная команда
K1 (4 разряда)	от 0 до 15
K2 (8 разрядов)	от 0 до 255
K3 (12 разрядов)	от 0 до 4095
K4 (16 разрядов)	от -32768 до 32767

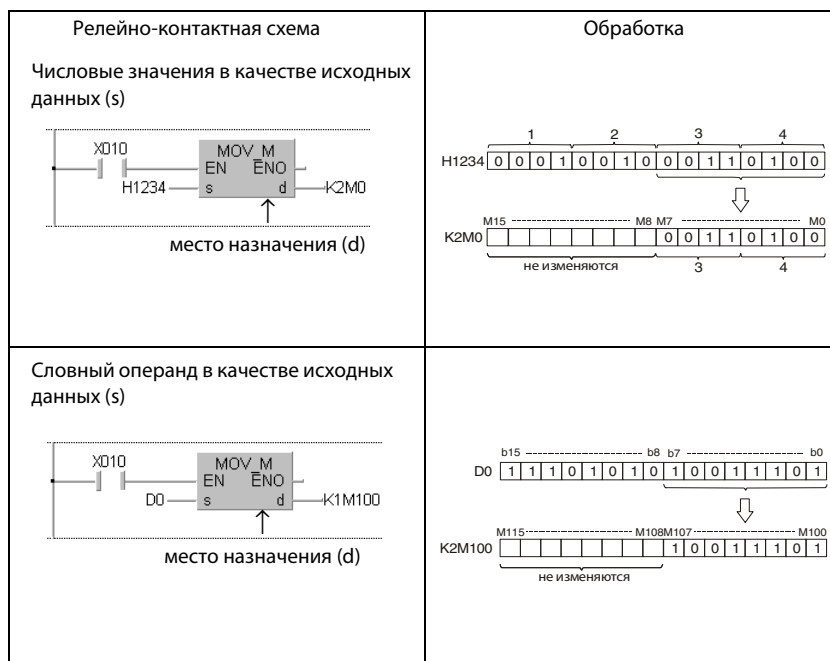
Неиспользуемые битовые адреса устанавливаются на ноль



**ПРИМЕЧАНИЕ** При поблочной адресации битовых операндов в качестве адреса первого битового операнда (начального адреса) можно указать любое значение.

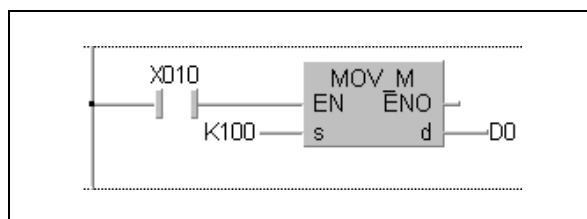
● Указание битовых блоков для d

Задание длины блока для целых данных d устанавливает диапазон адресов, в которые должны быть записаны данные. Битовые адреса выше указанных не учитываются.



● Применение словных операндов

Словные операнды задаются с помощью адресов, охватывающих 16 битов.

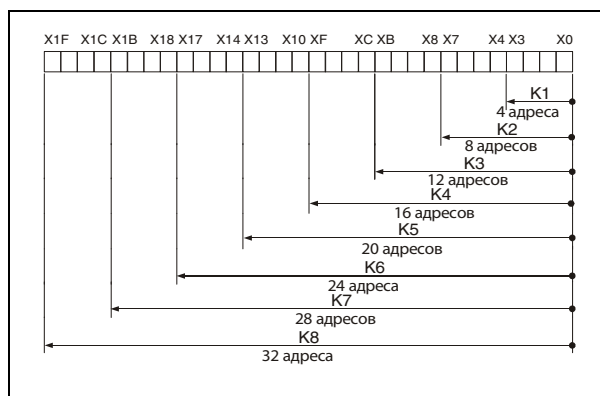


**Обработка данных типа "двойное слово" (32 бита)**

● Применение битовых операндов

Битовые операнды можно использовать для словных данных. Для этого необходимо задать количество битовых операндов (адресов). Возможна обработка до 32 битов в виде блоков по 4 бита. Длина блока устанавливается с помощью обозначений от K1 до K8.

- K1X0            4 адреса с X0 по X3
- K2X0            8 адресов с X0 по X7
- K3X0            12 адресов с X0 по XB
- K4X0            16 адресов с X0 по XF
- K5X0            20 адресов с X0 по X13
- K6X0            24 адреса с X0 по X17
- K7X0            28 адресов с X0 по X1B
- K8X0            32 адреса с X0 по X1F

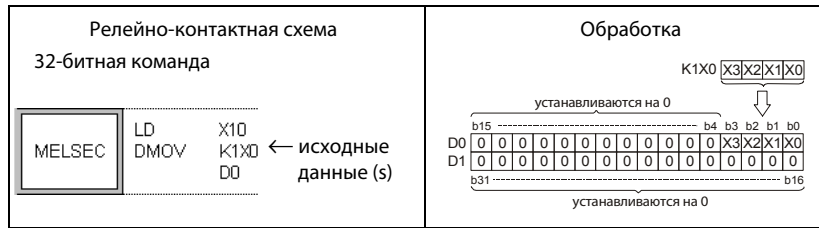


● Указание битовых блоков для s

В следующей таблице указан диапазон исходных данных, если для обозначения исходных данных s задается длина блока.

Длина блока	32-битная команда
K1 (4 разряда)	от 0 до 15
K2 (8 разрядов)	от 0 до 255
K3 (12 разрядов)	от 0 до 4095
K4 (16 разрядов)	от 0 до 65535
K5 (20 разрядов)	от 0 до 1048575
K6 (24 разряда)	от 0 до 16777215
K7 (28 разрядов)	от 0 до 268435455
K8 (32 разряда)	от -2147483648 до 2147483647

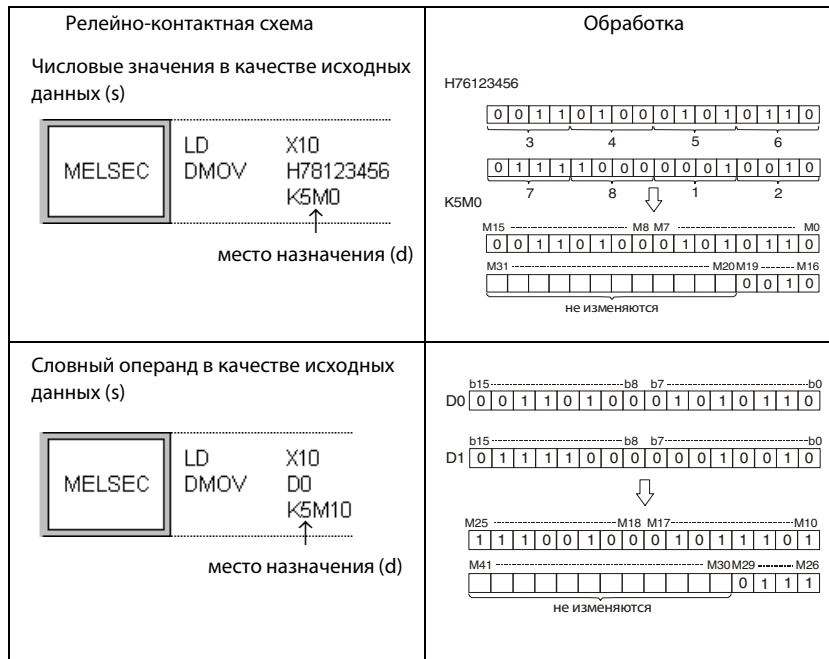
Неиспользуемые битовые адреса устанавливаются на ноль.



**ПРИМЕЧАНИЕ** При поблочной адресации битовых операндов в качестве адреса первого битового операнда (начального адреса) можно указать любое значение.

- Указание битовых блоков для d

Задание длины блока для целых данных (d) устанавливает диапазон адресов, в которые должны быть записаны данные. Битовые адреса выше указанных не учитываются.

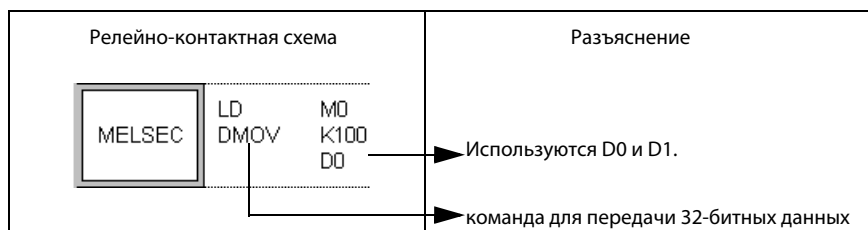


● Применение словных операндов

Двойные словные операнды охватывают два 16-битных операнда.

В зависимости от среды программирования и выбранного редактора, двойные словные операнды программируются по-разному.

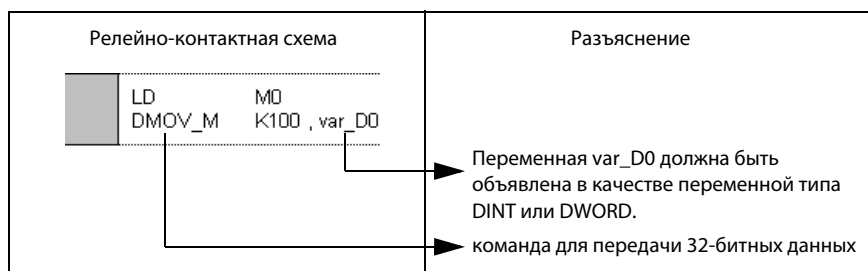
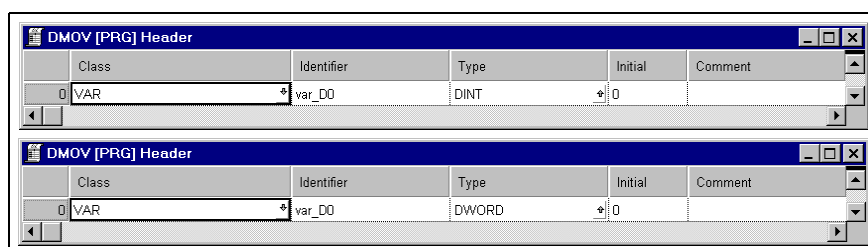
● В редакторе MELSEC среды GX IEC Developer



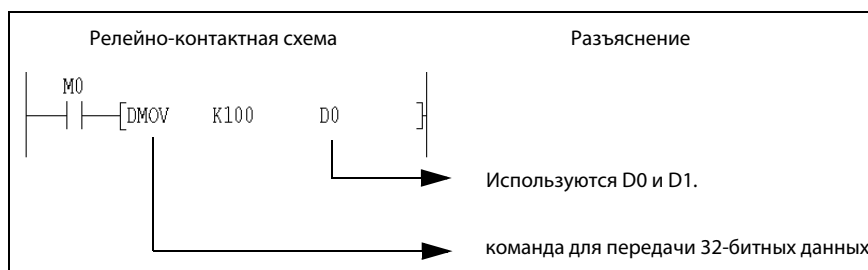
● В редакторе IEC среды GX IEC Developer

Чтобы 32-битный операнд можно было применять в редакторе IEC среды GX IEC Developer, переменная должна быть объявлена в заголовке программного компонента (POU).

Типы данных DWORD и DINT являются 32-битными.



● В редакторе GX Developer





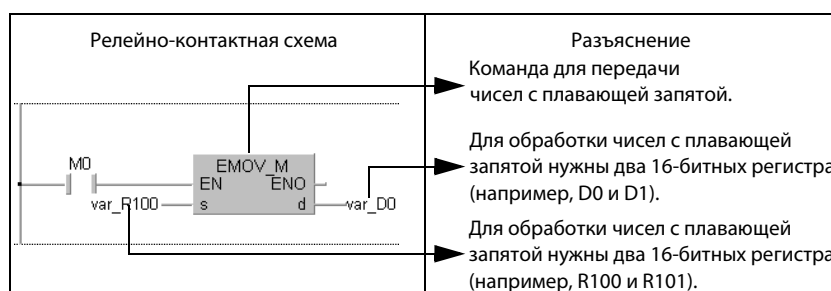
### Обработка данных типа REAL

Данные типа REAL являются 32-битными числами с плавающей запятой.

Сохранять числа с плавающей запятой возможно только в словных операндах.

Операнды, представляющие в командах числа с плавающей запятой, адресуются с помощью 16 младших битов. Сохраняемое 32-битное число с плавающей запятой записывается в два следующих друг за другом 16-битных регистра.

Если данными типа REAL требуется оперировать в процессорах AnA/AnU, то для этого необходимо использовать соответствующие специализированные команды (см. раздел 3.3 "Программирование специализированных команд").

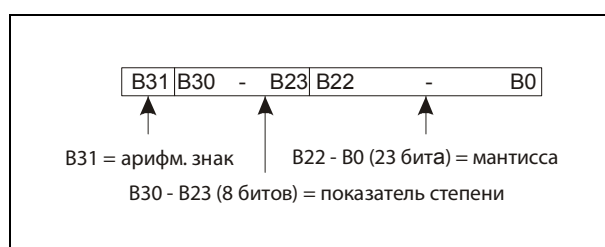


**ПРИМЕЧАНИЕ** В среде GX IEC Developer для чисел с плавающей запятой используется обозначение  $E\Box$ . Названия команд, оперирующих числами с плавающей запятой, начинаются с буквы E.

Для сохранения числа с плавающей запятой нужны два словных операнда. Для этого число раскладывается на следующие составные части:

арифметический знак;  $2^{[\text{показатель степени}]}$ ; [мантисса]

Конфигурация битов регистров и их значение разъясняются на следующей иллюстрации.



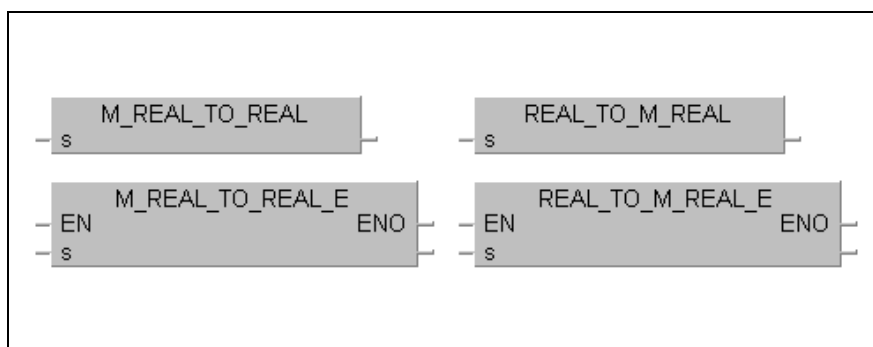
- Арифм. знак числа с плав. запятой: арифметический знак сохраняется в b 31.  
0 = положительный  
1 = отрицательный
- Показатель степени: n (из формулы  $2^n$ ) сохраняется в двоичном виде в битах с b23 по b30. Двоичное значение n проиллюстрировано ниже.

b23 ... b30	FFH	FEH	FDH			81H	80H	7FH	7EH			02H	01H	00H
n	своб.	127	126			2	1	0	-1			-125	-126	своб.

Пример: если в битах с b23 по b30 в двоичной кодировке записано число 81H, то n = 2.

- Мантисса: в 23 битах с b0 по b22 можно в двоичном виде отобразить 7 разрядов (XXXXXX или 1,XXXXXX).

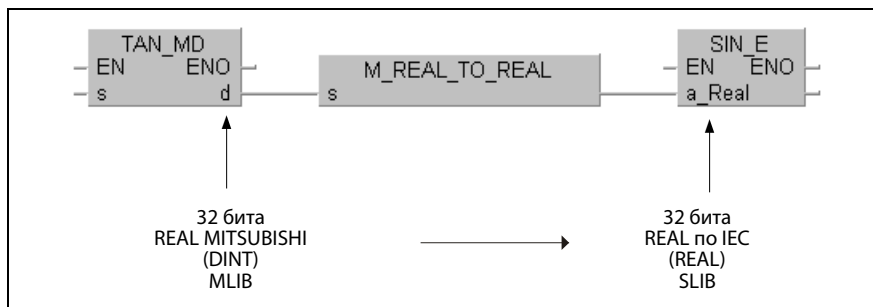
Так как в языке по стандарту IEC функция REAL использует входные и выходные данные типа REAL, а команды MELSEC – данные типа DINT, предусмотрены следующие вспомогательные функции для компенсации этого различия.



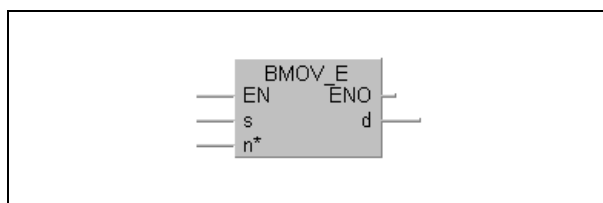
Преобразование типа данных REAL, соответствующего стандарту IEC, в тип данных, принятый для MELSEC, осуществляется командой REAL\_TO\_M\_REAL (REAL\_TO\_M\_REAL\_E).

Преобразование типа данных REAL, принятого в MELSEC, в тип данных, соответствующий стандарту IEC, осуществляет команда M\_REAL\_TO\_REAL (M\_REAL\_TO\_REAL\_E).

Пример: при использовании специализированных команд, оперирующих данными типа REAL, и команд по стандарту IEC необходимо преобразование "REAL to REAL".



При программировании в GX IEC Developer можно использовать команду BMOV\_E, выключающую проверку переменных. Эта команда не вырабатывает никакого дополнительного кода в программе контроллера.



В s можно указать любой тип данных, в том числе и массив. В n устанавливается, какое количество 16-битной информации копируется.

### 3.5.2 Адресация массивов и регистров в GX IEC Developer

#### Адресация 32-битных регистров

Для адресации 32-битных регистров (тип данных DINT, DWORD) необходимо объявить переменную в заголовке программного компонента (POU).

В следующем примере для перемещения 32-битного слова данных команда DMOV нуждается в двух 16-битных регистрах. При адресации в GX Developer и в редакторе MELSEC GX IEC Developer указывается только начальный регистр (в примере: D10, D20). Второй требуемый 16-битовый регистр (D11, D21) компилятор выделяет автоматически.

В редакторе IEC среды GX IEC Developer вместо указания начального регистра необходимо в заголовке программного компонента (POU) объявить переменные (в примере: var\_D10, var\_D20) определенного типа (в примере: DINT(32 бита)) в соответствии с заголовком команды. Для этих переменных компилятор выделяет адреса автоматически.

<p>Язык IL для MELSEC</p> <pre style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> MELSEC LD      X0       DMOV     D10            D20                 </pre>	<p>Релейно-контактная схема (язык LD)</p>	<p>Язык IL по МЭК</p> <pre style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> LD      X0 DMOV_M  var_D10 , var_D20                 </pre>																		
<p>Заголовок команды DMOV</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th>Class</th> <th>Identifier</th> <th>Type</th> <th>Initial</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>VAR_INPUT</td> <td>s</td> <td>ANY32</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>VAR_OUTPUT</td> <td>d</td> <td>ANY32</td> <td>0</td> <td></td> </tr> </tbody> </table>				Class	Identifier	Type	Initial	Comment	0	VAR_INPUT	s	ANY32	0		1	VAR_OUTPUT	d	ANY32	0	
	Class	Identifier	Type	Initial	Comment															
0	VAR_INPUT	s	ANY32	0																
1	VAR_OUTPUT	d	ANY32	0																
<p>Заголовок программного компонента (POU)</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th>Class</th> <th>Identifier</th> <th>Type</th> <th>Initial</th> <th>Comment</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>VAR</td> <td>var_D10</td> <td>DINT</td> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>VAR</td> <td>var_D20</td> <td>DINT</td> <td>0</td> <td></td> </tr> </tbody> </table>				Class	Identifier	Type	Initial	Comment	0	VAR	var_D10	DINT	0		1	VAR	var_D20	DINT	0	
	Class	Identifier	Type	Initial	Comment															
0	VAR	var_D10	DINT	0																
1	VAR	var_D20	DINT	0																

### Адресация массивов

При программировании команд, которые в качестве входного или выходного операнда (16-битный регистр) используют массив, в заголовке программного компонента (POU) необходимо объявить переменную в соответствии с заголовком команды.

Отдельные элементы массива адресуются путем указания массива и элемента массива в квадратных скобках (var\_xx[x]).

На следующей иллюстрации адресация с помощью массива пояснена на примере команды позиционирования поворотных столов (ROTC).

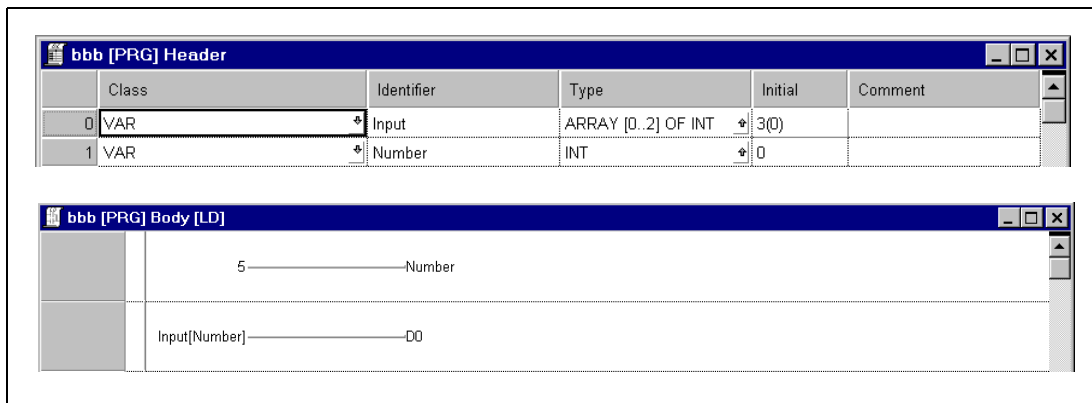
<p style="text-align: center;">Язык IL для MELSEC</p> <pre style="font-family: monospace; font-size: 0.9em;"> MELSEC LD      X0 OUT     M0 LD      X1 OUT     M1 LD      X2 OUT     M2 LD      X10 ROT     D200         K10         K2         M0                 </pre>	<p style="text-align: center;">Релейно-контактная схема (язык LD)</p>	<p style="text-align: center;">Язык IL по МЭК</p> <pre style="font-family: monospace; font-size: 0.9em;"> LD      X0 ST      var_M0[0] LD      X1 ST      var_M0[1] LD      X2 ST      var_M0[2] LD      X10 ROT     var_D200, 10, 2, var_M0                 </pre>																														
<p>Заголовок команды ROTC</p> <table border="1" style="width: 100%; border-collapse: collapse; font-size: 0.8em;"> <thead> <tr> <th style="width: 5%;"></th> <th style="width: 25%;">Class</th> <th style="width: 15%;">Identifier</th> <th style="width: 25%;">Type</th> <th style="width: 10%;">Initial</th> <th style="width: 20%;">Comment</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>VAR_INPUT</td> <td>s</td> <td>ARRAY [ 1.. 3 ] OF ANY16</td> <td>0,0,0</td> <td></td> </tr> <tr> <td>1</td> <td>VAR_INPUT</td> <td>n1</td> <td>ANY16</td> <td>0</td> <td></td> </tr> <tr> <td>2</td> <td>VAR_INPUT</td> <td>n2</td> <td>ANY16</td> <td>0</td> <td></td> </tr> <tr> <td>3</td> <td>VAR_OUTPUT</td> <td>d</td> <td>ARRAY [ 1.. 8 ] OF BOOL</td> <td>8(FALSE)</td> <td></td> </tr> </tbody> </table>				Class	Identifier	Type	Initial	Comment	0	VAR_INPUT	s	ARRAY [ 1.. 3 ] OF ANY16	0,0,0		1	VAR_INPUT	n1	ANY16	0		2	VAR_INPUT	n2	ANY16	0		3	VAR_OUTPUT	d	ARRAY [ 1.. 8 ] OF BOOL	8(FALSE)	
	Class	Identifier	Type	Initial	Comment																											
0	VAR_INPUT	s	ARRAY [ 1.. 3 ] OF ANY16	0,0,0																												
1	VAR_INPUT	n1	ANY16	0																												
2	VAR_INPUT	n2	ANY16	0																												
3	VAR_OUTPUT	d	ARRAY [ 1.. 8 ] OF BOOL	8(FALSE)																												
<p>Заголовок программного компонента (POU)</p> <table border="1" style="width: 100%; border-collapse: collapse; font-size: 0.8em;"> <thead> <tr> <th style="width: 5%;"></th> <th style="width: 25%;">Class</th> <th style="width: 15%;">Identifier</th> <th style="width: 25%;">Type</th> <th style="width: 10%;">Initial</th> <th style="width: 20%;">Comment</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>VAR</td> <td>var_D200</td> <td>ARRAY [0..2] OF INT</td> <td>3(0)</td> <td></td> </tr> <tr> <td>1</td> <td>VAR</td> <td>var_M0</td> <td>ARRAY [0..7] OF BOOL</td> <td>8(FALSE)</td> <td></td> </tr> </tbody> </table>				Class	Identifier	Type	Initial	Comment	0	VAR	var_D200	ARRAY [0..2] OF INT	3(0)		1	VAR	var_M0	ARRAY [0..7] OF BOOL	8(FALSE)													
	Class	Identifier	Type	Initial	Comment																											
0	VAR	var_D200	ARRAY [0..2] OF INT	3(0)																												
1	VAR	var_M0	ARRAY [0..7] OF BOOL	8(FALSE)																												

В заголовке команды ROTC указано, что область входных операндов s состоит из 3 элементов массива типа ANY16, а область выходных операндов – из 8 элементов массива типа BOOL.

В GX Developer, в редакторе MELSEC среды GX IEC Developer и в MELSEC MEDOC в качестве областей входных и выходных операндов s и d указываются только начальные операнды D200 и M0. Компилятор выделяет для s регистры с D200 по D202, а для d – маркеры с M0 по M7.

В редакторах по стандарту МЭК (IEC) для s и d требуется определять массивы. Входной массив s определен в виде var\_D200. Он состоит из 3 элементов (var\_D200[0] – var\_D200[2]) типа INT (16-битные целые числа). Выходной массив d определен в виде var\_M0. Он состоит из 8 элементов (var\_M0[0] – var\_M0[7]) типа BOOL (один бит). Для этих переменных компилятор выделяет адреса автоматически.

**ПРИМЕЧАНИЕ** При адресации массивов можно использовать и переменные. В этом случае вместо указания элемента массива в квадратных скобках вводится какой-либо идентификатор, например, [Noter]. Переменная "Noter" должна быть объявлена в заголовке программного компонента (POU). Впоследствии в регистр "Noter" можно поместить значение, соответствующее элементу массива.



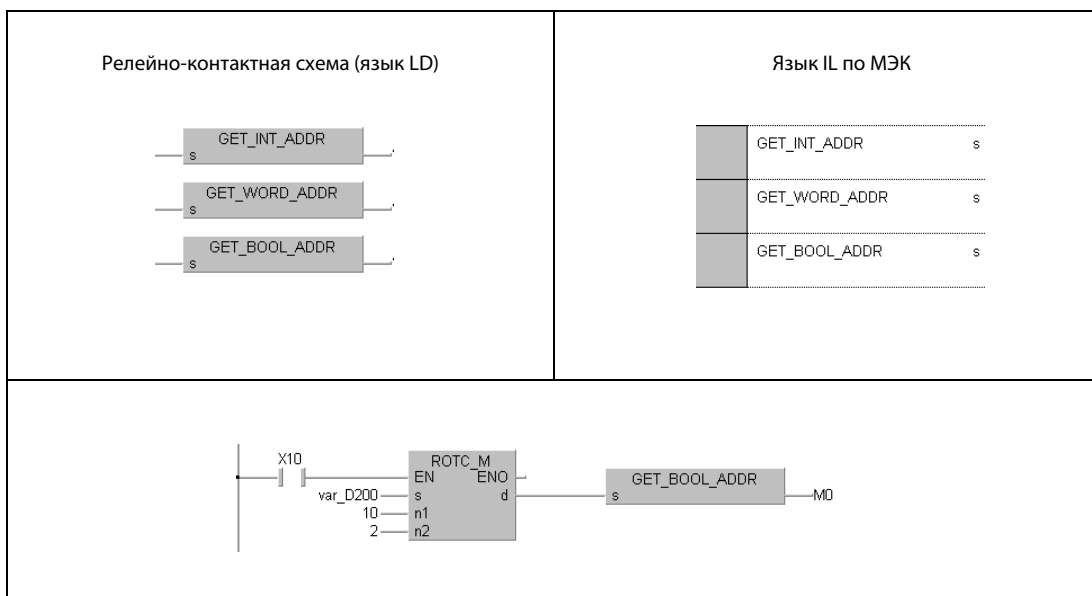
### Команды преобразования массива в начальный адрес

Набор команд для конвертации выходного массива в начальный адрес области операндов состоит из трех команд.

Команда GET\_INT\_ADDR преобразует выходной массив с элементами типа INT (16-битные целые числа) в начальный адрес области операндов.

Команда GET\_WORD\_ADDR преобразует выходной массив с элементами типа WORD (16-битные слова) в начальный адрес области операндов.

Команда GET\_BOOL\_ADDR преобразует выходной массив с элементами типа BOOL (отдельные биты) в начальный адрес области операндов.



После преобразования элементы массива можно обрабатывать в качестве отдельных операндов. Благодаря этому отпадает необходимость в объявлении переменной в заголовке программного компонента (POU).

В вышеприведенном примере программы с командой ROTC, выполнив преобразование, можно вместо элементов массива var\_M0[0] – var\_M0[7] использовать маркеры с M0 по M7.

Операнды конвертированных элементов массива в средах GX IEC Developer и GX Developer адресуются одинаково.

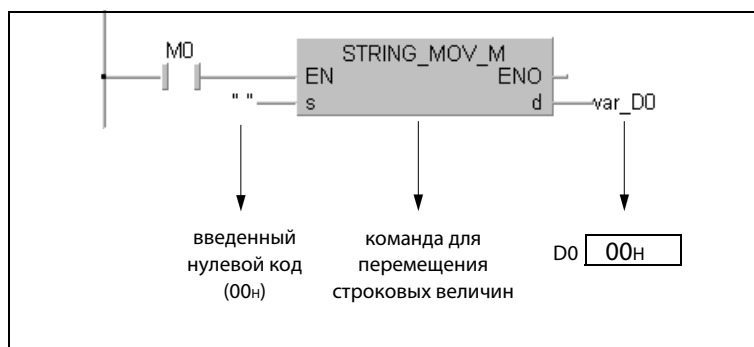
Эти команды преобразуют только выходные массивы. Входные массивы требуется адресовать и объявлять, как это было описано выше.

### 3.5.3 Применение строковых данных (STRING)

Данные типа STRING (\$) оперируют строковыми величинами. Строковыми величинами являются все введенные знаки (макс. 50 знаков) до нулевого кода (00н).

- Если среди знаков введен нулевой код (00н)

Для сохранения нулевого кода требуется слово данных (регистр).

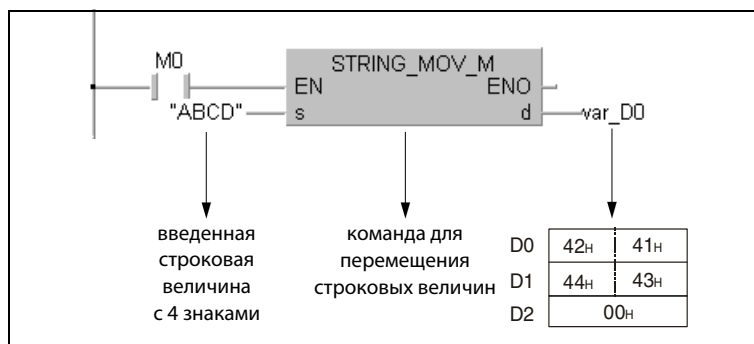


- Если число знаков в строковой величине четно

Чтобы сохранить строковую величину с четным количеством знаков, требуется количество слов данных (регистров), рассчитываемое по следующей формуле:

$$(число\ знаков/2) + 1$$

Например, если строковую величину "ABCD" требуется записать по адресу D0, то для строковой величины нужны регистры с D0 по D1, а для нулевого кода, обозначающего конец строковой величины, нужен регистр D2.

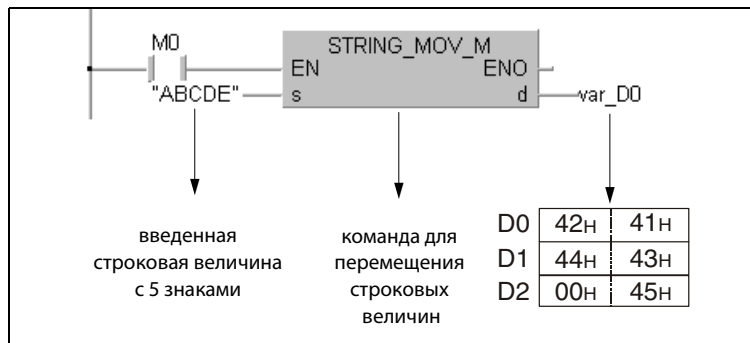


- Если число знаков в строковой величине нечетное

Чтобы сохранить строковую величину с нечетным количеством знаков, требуется количество слов данных (регистров), рассчитываемое по следующей формуле:

*(число знаков/2)*

Например, если строковую величину "ABCDE" требуется записать по адресу D0, то для нее нужны регистры с D0 по D2. Нулевой код, обозначающий конец строковой величины, записывается в старший байт регистра D2.



## 3.6 Индексация

Так как индексация в процессорах "System Q" и серии "Q" отличается от индексации в процессорах серии "A", в нижеследующих разделах 3.6.1 и 3.6.2 более подробно разъяснены особенности центральных процессоров различных типов.

Индексация используется для косвенной адресации операндов посредством индексного регистра. Если в программе используются индексы, то в качестве адреса операнду присваивается непосредственно введенный адрес операнда плюс содержимое индексного регистра.

### Применение индексов в программе

На следующей иллюстрации показан пример программы с индексацией. В первой строке программы индексному регистру Z0 присваивается значение 1. Этот регистр используется во второй строке программы в качестве индекса для D10. Поэтому в D0 сохраняется значение из D11 ( $D10Z = D(10 + 1) = D11$ ).

Релейно-контактная схема	Разъяснение
	В индексном регистре Z0 сохраняется константа 1.
	Данные из регистра ( $D10 + Z0(1) = D11$ ), индексированного с помощью Z0, сохраняются в D0.

На следующей иллюстрации приведен еще один пример индексации для разъяснения обработки операндов ( $Z0 = 20, Z1 = 5$ ).

Релейно-контактная схема	Разъяснение
	В индексном регистре Z0 сохраняется константа 20.
	В индексном регистре Z1 сохраняется константа 5.
	Константа 100 индексирована на основе Z0 ( $100 + Z0(20) = 120$ ) и записывается в регистр W53 ( $W53 + Z1(5) = W58$ ), индексированный на основе Z1.



**Операнды, которые можно адресовать с использованием индексов**

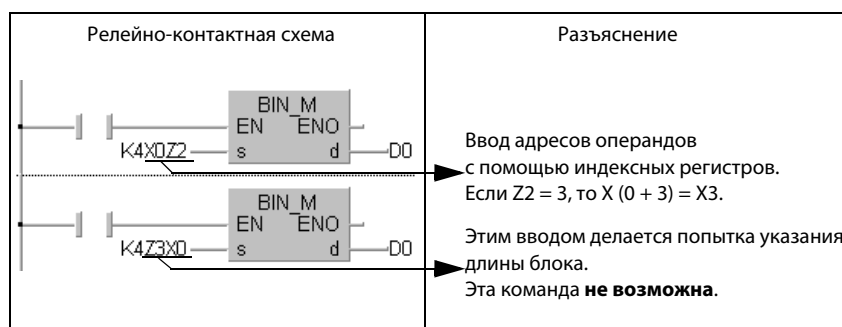
Индексирование можно применять для операндов, контактов и катушек. Индексные регистры служат для косвенной адресации операндов и содержат числовые значения в диапазоне между -32768 и 32767.

**Операнды, которые невозможно адресовать с использованием индексов**

Операнд	Значение
E	числа с плавающей запятой
\$	строковые величины
□.□	битовая адресация словных операндов
FX, FY, FD	функциональные операнды
P	указатели, используемые в качестве меток
I	указатели прерывания, используемые в качестве меток
Z	индексные регистры
S	шаговые маркеры
TV, STV	заданные значения таймеров
CV	заданные значения счетчиков
N	вложенные уровни
A0	сумматор
A1	сумматор

**Битовые данные (кроме процессоров AnN)**

При адресации блоков также можно использовать индексацию. Однако влиять на длину блоков с помощью индексных регистров невозможно.



### 3.6.1 Особенности процессоров Q и QnA

Процессоры Q и QnA имеют 16 индексных регистров (Z0–Z15). В следующей таблице указаны диапазоны значений таймеров и счетчиков, которые можно адресовать с помощью индексов.

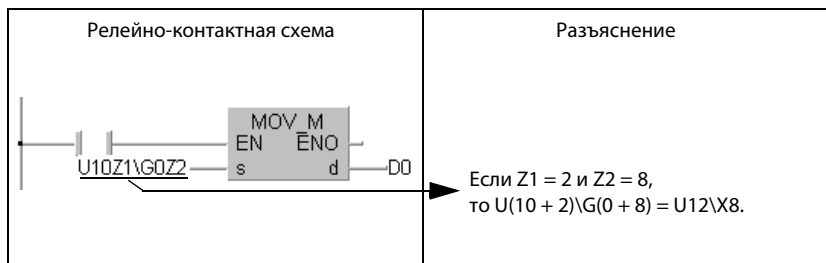
Операнд	Значение	Пример применения
TC	Для адресации катушек и контактов таймеров можно использовать только регистры Z0 и Z1.	
CC	Для адресации катушек и контактов счетчиков можно использовать только регистры Z0 и Z1.	

**ПРИМЕЧАНИЕ** В отношении индексированной адресации фактических значений таймеров и счетчиков никаких ограничений не имеется.

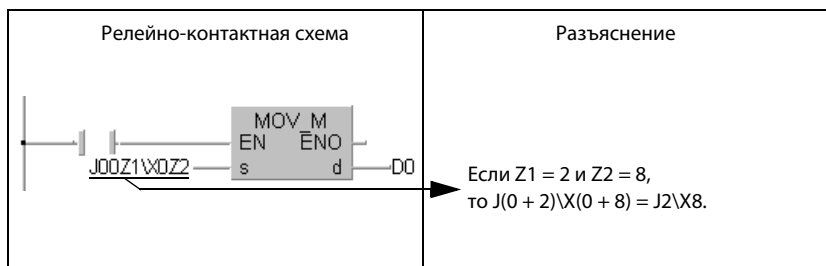
Релейно-контактная схема	Разъяснение
	<p>заданное значение таймера (TV)                      Индексация <b>не возможна</b>.</p>
	<p>фактическое значение таймера (TN)                      Индексация возможна.</p>
	<p>заданное значение счетчика (CV)                      Индекса <b>не возможна</b>.</p>
	<p>фактическое значение счетчика (CN)                      Индексация возможна.</p>

Еще одно отличие от процессоров "А" заключается в возможности индексированного указания адресов входов/выходов и адресов буферной памяти, номеров коммуникационных сетей и адресов операндов модулей сетевой коммуникации.

На рисунке ниже показана адресация входов/выходов и адресов буферной памяти в специальных модулях.



На рисунке ниже показана адресация номеров коммуникационных сетей и операндов в модулях сетевой коммуникации.



**ПРИМЕЧАНИЕ** Более подробная информация о специальных модулях и модулях сетевой коммуникации имеется в следующих руководствах: "QnA CPU-Programming Manual (Fundamentals)", "QCPU (Q mode) Users Manual (Functions/programming fundamentals)", а также в руководствах, посвященных конкретным устройствам.

### 3.6.2 Особенности процессоров AnA, AnAS и AnU

Адреса операндов в программе можно индексировать (индексами Z или V). Для различения индексов предусмотрен арифметический знак.

В следующих случаях при обработке команд возникает ошибка обработки.

- При индексации превышен диапазон адресов операндов  
Константы K и H в этом случае игнорируются.
- При индексации начальный адрес области операндов превышает допустимый диапазон адресов.

**ПРИМЕЧАНИЯ** *Для сокращения времени обработки процессоры серий AnA, AnAS и AnU при индексации не проверяют адреса операндов. По этой причине ошибки, связанные с индексацией, не распознаются в качестве ошибок обработки.*

*Если в связи с индексацией возникла ошибка, то данные операндов могут самопроизвольно измениться.*

*Поэтому программы, содержащие индексацию, следует писать с максимальной внимательностью!*

В программах для процессоров AnA, AnAS или AnU индексировать можно и битовые операнды, используемые в командах LD, OUT или подобных командах.

#### Сохранение 32-битных данных в индексных регистрах

32-битные данные можно сохранять в расширенных индексных регистрах (с Z1 по Z6 и с V1 по V6) процессоров AnA или AnU. Для этого необходимо парами использовать следующие индексные регистры:

Z1 и V1

Z2 и V2

Z3 и V3

Z4 и V4

Z5 и V5

Z6 и V6

Zn содержит младшие 16 битов, Vn содержит старшие 16 битов. В 32-битных командах разрешается указывать только операнд Z. Если указан операнд V, выполнение программы не возможно.

32-битные команды можно сохранять только в вышеуказанных парах регистров. Иные комбинации не допускаются. Если операнд пары регистров используется в какой-либо команде для индексации, то содержащиеся в этом регистре данные обрабатываются в качестве 16-битных данных индексации.

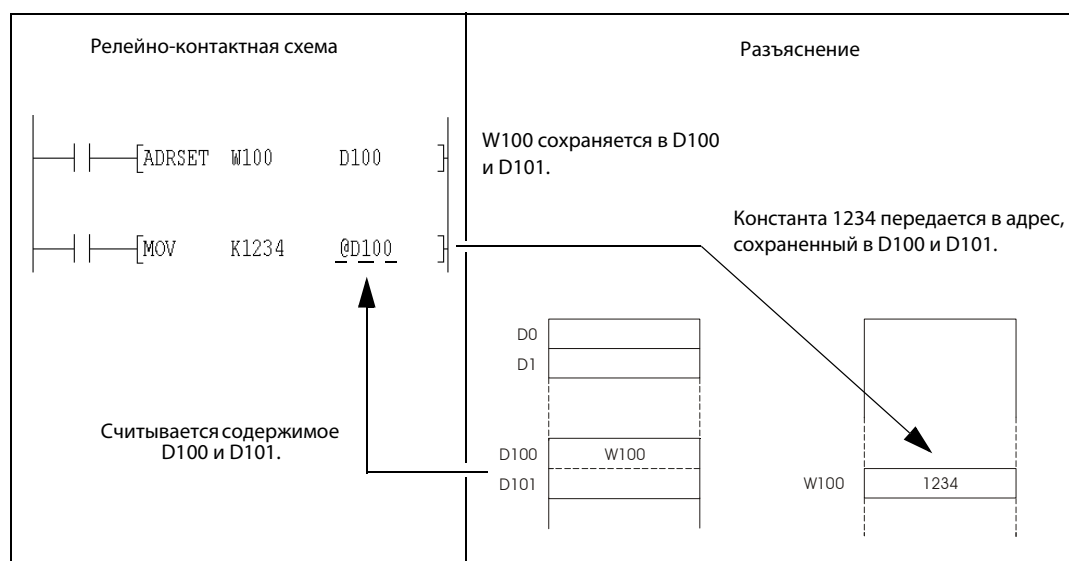
### 3.7 Косвенная адресация (только в GX Developer)

При косвенной адресации адрес операнда записывается в словный операнд. Программа обращается не непосредственно к операнду, в отношении которого должна выполняться операция, а через операнд, содержащий сохраненный адрес. Косвенную адресацию можно применять в случае, если индексации недостаточно.

В программе операнд, который содержит адрес косвенного адресуемого операнда, обозначается знаком "@". Например, при указании "@D100" содержимое регистров D100 и D101 используется в качестве адреса.

Адрес косвенно адресуемого операнда сохраняется с помощью команды ADRSET.

**ПРИМЕЧАНИЕ** Команда ADRSET не может использоваться при программировании в среде GX IEC Developer.

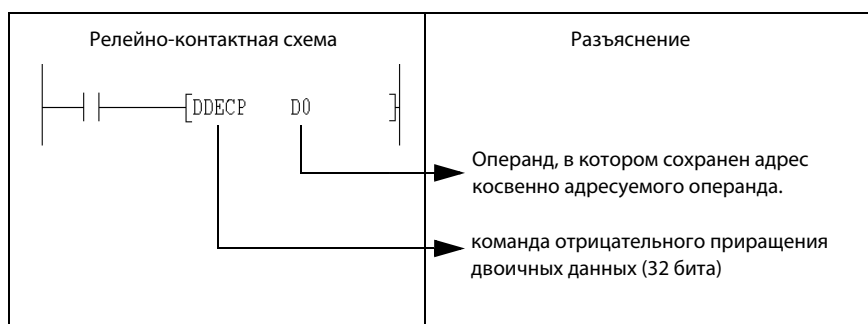
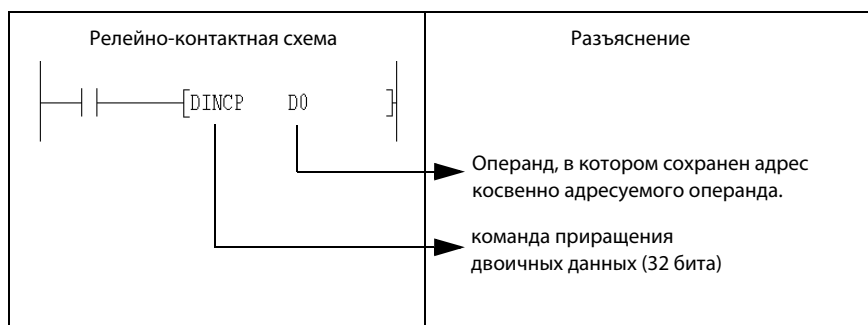


В следующей таблице перечислены операнды, которые можно адресовать косвенно.

Операнд		Косвенная адресация	Пример косвенной адресации
Внутренние операнды (системные, пользовательские)	битовые	не возможна	—
	словные	возможна	<ul style="list-style-type: none"> <li>• @D100</li> <li>• @D100Z2 (индексация)</li> </ul>
MELSECNET/10	битовые	не возможна	—
	словные	возможна (адрес операнда не может быть сохранен с помощью команды ADRSET)	<ul style="list-style-type: none"> <li>• @J1\W10</li> <li>• @J1Z1\W10Z2 (индексация)</li> </ul>
Специальные модули			<ul style="list-style-type: none"> <li>• @U10\G0</li> <li>• @U10Z1\G0Z2 (индексация)</li> </ul>
Индексные регистры Zn		не возможна	—
Регистры файлов		возможна	<ul style="list-style-type: none"> <li>• @R0, @ZR20000</li> <li>• @R0Z1, @ZR20000Z1 (индексация)</li> </ul>
Вложенные уровни		не возможна	—
Указатели			—
Константы			—
Прочие			—

**ПРИМЕЧАНИЕ** Более подробную информацию на тему "Операнды" можно найти в руководстве "QnA CPU-Programming Manual (Fundamentals)" или "QCPU (Q mode) Users Manual (Functions/programming fundamentals)".

**ПРИМЕЧАНИЕ** Для сохранения адреса косвенно адресуемого операнда используются два слова. Поэтому, если сохраненный адрес требуется увеличить или уменьшить путем вычислительных операций, требуется складывать или вычитать 32-битные данные. В следующем примере операнд, содержащий адрес косвенно адресуемого операнда, получает приращение и уменьшается с помощью 32-битных команд. В результате этих операций адрес косвенно адресуемого операнда повышается или понижается на 1.



## 3.8 Ошибки обработки

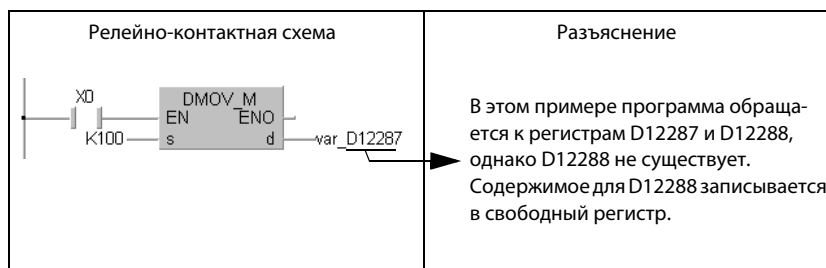
Ошибки обработки возникают в следующих случаях:

- Если выполняются условия ошибки, названные в пункте "Источники ошибок" при описании отдельных команд, вырабатывается код ошибки.
- Если при использовании буферного регистра к predetermined адресу ввода-вывода не подключен специальный модуль.
- Если при использовании операндов связи отсутствует соответствующая сеть.
- Если при использовании операндов связи не подключен модуль сетевой коммуникации с predetermined адресом ввода-вывода.

**ПРИМЕЧАНИЕ** Если в параметрах регистров файлов была определена карта памяти (только в случае процессоров Q/QnA), однако эта карта не установлена, возникает сообщение об ошибке (2401 = ошибка набора файлов).  
Если осуществлено обращение к регистру файлов, хотя в параметрах регистры файлов не были определены, никакой код ошибки не вырабатывается. При считывании регистра файлов появляется код "FFFFH".

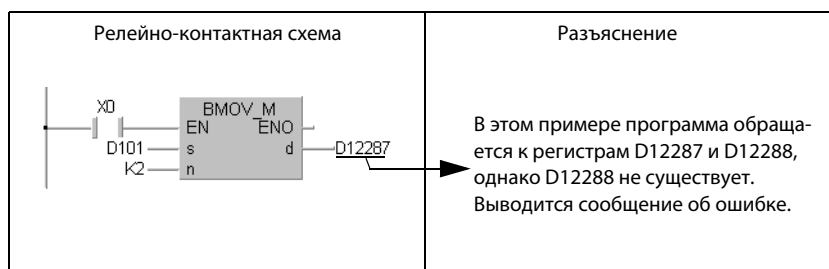
### 3.8.1 Проверка области операндов

- Если команды используют операнды фиксированной длины (MOV, DMOV и т. п.), область операндов не проверяется.  
В этих случаях при превышении диапазона адресов данные записываются в свободный регистр. Например, если присвоены 12k адресов, сообщение об ошибке не появляется до тех пор, пока не будет превышен адрес регистра D12287.



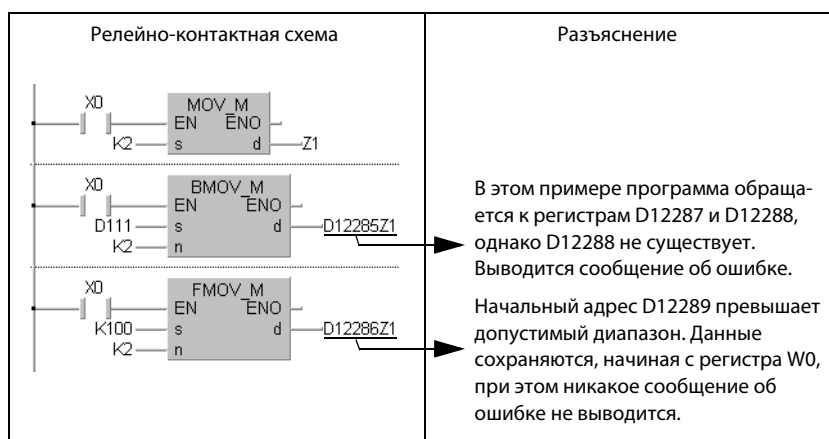
Область операндов не проверяется и в случае индексированной адресации.

- Если команды используют операнды переменной длины (BMOV, FMOV и другие команды, использующие начальные адреса), выполняется проверка области операндов. Если диапазон адресов превышает, возникает сообщение об ошибке. Например, если присваиваются 12k адресов, то сообщение об ошибке возникает лишь при превышении адреса регистра D12287.

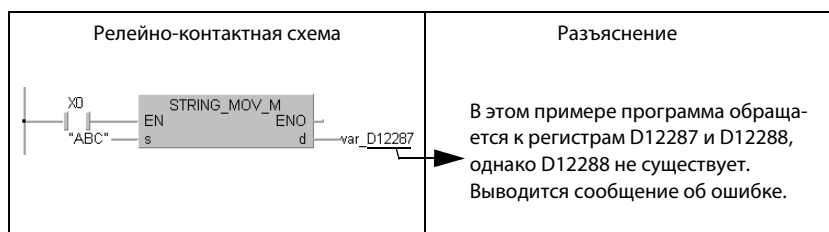


Область операндов проверяется и при индексируемой адресации.

Если начальный адрес операнда превышает диапазон адресов, сообщение об ошибке не возникает.



- Так как строковые величины имеют переменную длину, выполняется проверка области операндов. Если соответствующая область операндов превышает, вырабатывается сообщение об ошибке. Например, если присваиваются 12k адресов, сообщение об ошибке не появляется до тех пор, пока не будет превышен адрес регистра D12287.



- При индексированной адресации непосредственного вывода (DY) проверка области операндов выполняется.



### 3.8.2 Проверка данных операндов

#### При использовании двоичных данных

- Если результат обработки превышает диапазон, никакое сообщение об ошибке не возникает. Флаг переноса (переполнения) в этом случае не устанавливается.

#### При использовании двоично-десятичных данных

- Проверяется каждый знак двоично-десятичных значений (от 0 до 9). Если отдельный знак превышает диапазон от 0 до 9 (т. е. имеет значение от A до F), возвращается код ошибки.
- Если результат обработки превышает диапазон, никакое сообщение об ошибке не возникает. Флаг переноса (переполнения) в этом случае не устанавливается.

#### При использовании чисел с плавающей запятой

Ошибки обработки возникают в следующих случаях:

- Число с плавающей запятой принимает значение 0
- Абсолютное значение числа с плавающей запятой снижается ниже  $1.0 \times 2^{-127}$
- Абсолютное значение числа с плавающей запятой превышает  $1.0 \times 2^{129}$

#### При использовании строковых величин

Данные операндов не проверяются.

## 3.9 Условия выполнения команд

### 3.9.1 Входное условие

Возможны следующие четыре условия выполнения команд:

- Выполнение без условий

Команды выполняются без учета состояния сигнала операндов.  
Пример: LD X0, OUT Y10

- Выполнение при установленном условии

Команды выполняются до тех пор, пока условие выполнения находится в установленном состоянии.  
Пример: MOV, FROM

- Выполнение при положительном фронте

Команды выполняются при положительном фронте условия выполнения (т. е. при изменении сигнала с 0 на 1).  
Пример: PLS, MOVP

- Выполнение при отрицательном фронте

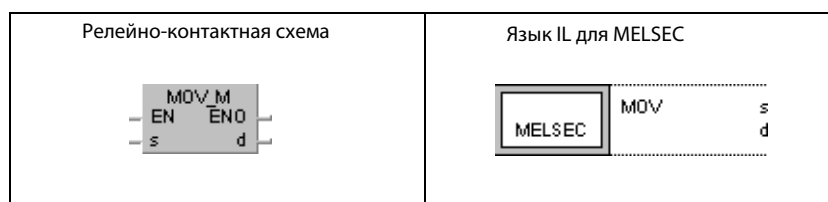
Команды выполняются при отрицательном фронте условия выполнения (т. е. при изменении сигнала с 1 на 0).  
Пример: PLF

Большинство команд выполняются при условиях следующих двух типов:

- при установленном условии выполнения
- при положительном фронте условия выполнения

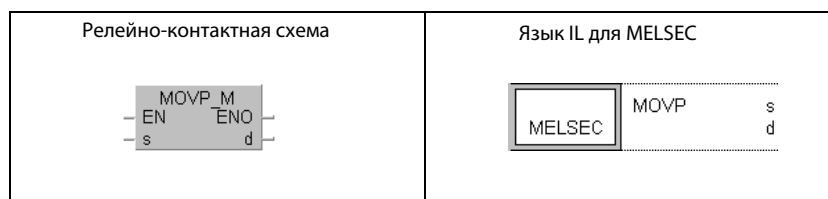
В первом из этих двух случаев команда выполняется до тех пор, пока условие выполнения установлено. Такие команды не имеют никакого особого обозначения.

Пример: MOV\_M/MOV

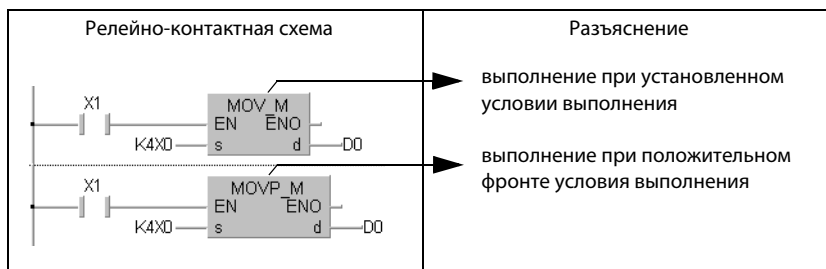


Во втором случае команда выполняется только при изменении сигнала с 0 на 1.

Пример: MOVP\_M/MOVP



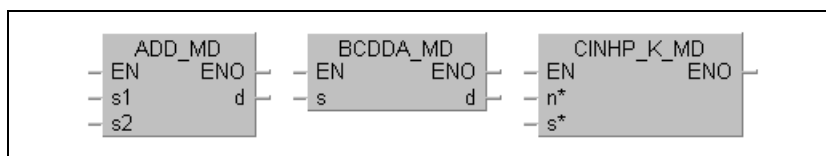
Следующий пример иллюстрирует выполнение команды MOV при установленном условии выполнения, а также при положительном фронте условия выполнения.



### 3.9.2 Вход EN и выход ENO

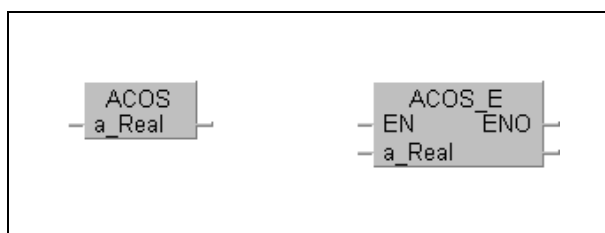
Все описываемые в этом руководстве команды входят в библиотеку изготовителя GX IEC Developer . Помимо входных и выходных переменных, эти команды всегда имеют вход EN и выход ENO.

На рисунке ниже изображены некоторые команды MELSEC из библиотеки изготовителя среды GX IEC Developer.



В стандартной библиотеке МЭК (IEC) почти все команды встречаются в двух видах, различающихся только окончанием "\_E". Команды с этим окончанием имеют вход EN и выход ENO.

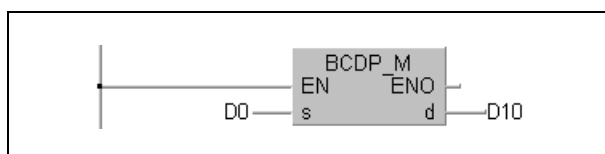
На рисунке ниже изображены две команды МЭК (IEC) из стандартной библиотеки среды GX IEC Developer.



На следующих примерах показано отличие в обработке команды с входом EN и выходом ENO по сравнению с командой без этого входа и выхода.

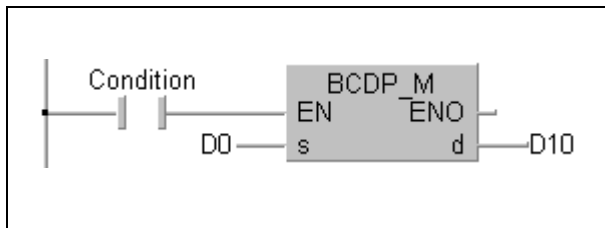
Пример 1: без дополнительного соединения

Без дополнительного соединения условие выполнения команды установлено постоянно.



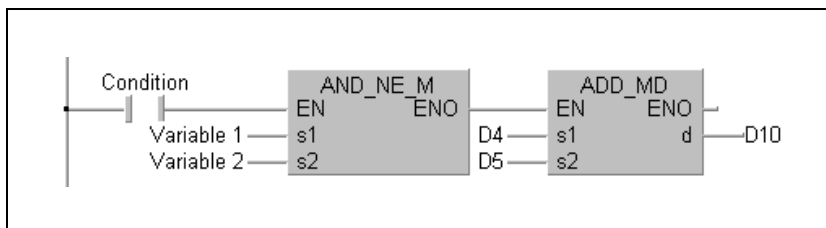
Пример 2: соединение с контактом

Если вход EN соединен с контактом, команда выполняется при выполняющемся условии.



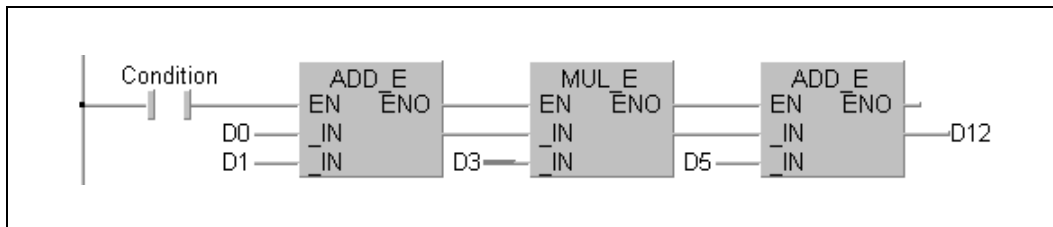
Пример 3: соединение с результатом операции

Если на вход EN подан булевый результат вычислительной операции, команда выполняется только при результате TRUE.



Пример 4: соединение с предыдущей командой

Если вход EN подключен к выходу ENO предыдущей команды, то эти команды выполняются только при выполняющемся условии.



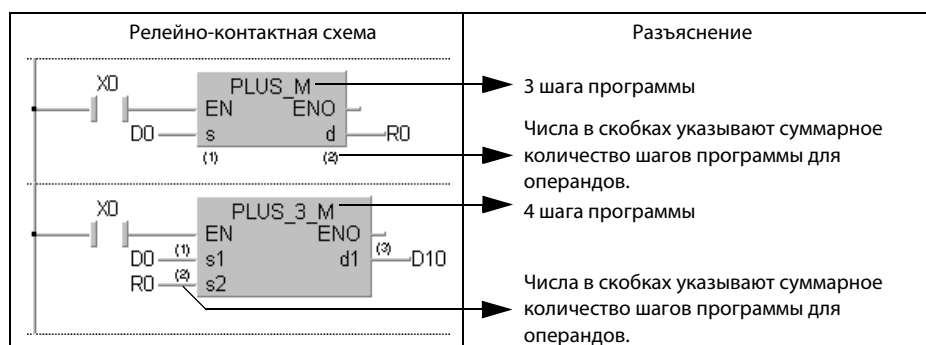
**ПРИМЕЧАНИЕ** Подключать выход ENO не обязательно. Сигнал на входе EN проводится насквозь к выходу ENO. Если вход EN имеет состояние "TRUE", то и выход ENO имеет состояние "TRUE".

## 3.10 Количество шагов программы

Чтобы не превышать область памяти, выделенную для шагов программы во внутренней памяти, а также в ROM или RAM на картах и кассетах памяти, необходимо рассчитывать общее количество шагов программы. В следующих разделах описывается расчет шагов команд для процессоров System Q, OnA и A.

### 3.10.1 В случае процессоров "System Q" и QnA

Количество шагов программы, расходуемых командой, зависит от количества базовых шагов. Для выполнения большинства команд нужно лишь определенное количество базовых шагов. Количество базовых шагов зависит от количества используемых операндов плюс 1. В следующем примере показан расчет количества базовых шагов для команды PLUS.



- Количество шагов программы в случае применения входных и выходных команд:

Количество шагов программы в случае применения входных команд (LD, LDI, AND, ANI, OR, ORI) зависит от используемых операндов.

Если используются внутренние операнды или регистры файлов (от R0 до R32767), это количество равно 1.

Если используются непосредственно адресуемые входы (DX), количество равно 2.

При использовании других операндов количество равно 3.

Количество шагов программы в случае применения выходных команд (LDP, LDF, ANDP, ANDF, ORP, ORF) зависит от используемых операндов.

При использовании внутренних операндов или регистров файлов (от R0 до R32767) количество равно 2.

При использовании непосредственно адресуемых входов (DX) количество равно 3.

При использовании прочих операндов количество равно 4.

- Количество шагов программы в случае применения некоторых команд передачи:

Операнды, увеличивающие количество шагов программы	Добавляющиеся шаги	Пример
Операнды специальных модулей	1	MOV <u>U4\G10</u> D0
Операнды связи		MOV <u>J3\B20</u> D0
Последовательно адресуемые регистры файлов		MOV <u>ZR123</u> D0
32-битные константы		DMOV <u>K123</u> D0
Числа с плавающей запятой в качестве констант		EMOV <u>E0.1</u> D0
Строковые величины	При нечетном количестве: (число знаков/2) – 1 При четном количестве: число знаков/2	\$MOV „123“ D0

Если имеют место сразу несколько факторов, количество шагов суммируется. Например, если запрограммирована команда MOV U1\G10 ZR123, то добавляется 1 шаг для буферной памяти и 1 шаг для последовательно адресуемого регистра файлов, в результате чего получается общее количество шагов 2.

### 3.10.2 В случае процессоров AnA, AnAS и AnU

При написании программ для процессоров AnA, AnAS или AnU необходимо соблюдать ряд особенностей, которым посвящен этот раздел.

Если команда обращается к одному из указанных в таблице адресов операндов (расширенная область серии AnA), количество шагов увеличивается на 1.

Операнд	Диапазон адресов
Маркеры M, L, S	от 2048 до 8191
Таймеры T	от 256 до 2047
Счетчики C	от 256 до 1023
Маркеры связи B	от 400 до FFF
Регистры данных D	от 1024 до 6143
Регистры связи W	от 400 до FFF

Если операнду из расширенной области адресов присваивается индекс с помощью расширенного индексного регистра, количество шагов также увеличивается на 1.

На следующей иллюстрации приведены некоторые примеры расчета шагов программы. В первом примере показаны шаги при использовании команд из обычного диапазона адресов.

В последующих примерах показаны шаги при использовании операндов из расширенного диапазона адресов.

Релейно-контактная схема	Разъяснение
	LD T0 1 шаг + D0 W010 <u>5 шагов</u> <b>6 шагов</b>
	LD T300 2 шага + D0 W800 <u>6 шагов</u> <b>8 шагов</b>
	LD T1000 2 шага + D2000 W010 Z1 <u>6 шагов</u> <b>9 шагов</b>
	LD T0 1 шаг + D2000 Z1 D300 <u>5 шагов</u> <b>6 шагов</b>

Если в одношаговой команде (например, LD или OUT) используется индексация, количество шагов увеличивается на 1.

В следующем примере показаны различия между программированием с индексами и без индексов. Количество шагов повышается только на 1 даже в том случае, если для индексации используется расширенный индексный регистр (от Z1 до Z6, от V1 до V6).

Релейно-контактная схема	Разъяснение
	LD X0 1 шаг + OUT Y40 <u>1 шаг</u> <b>2 шага</b>
	LD X0 Z 2 шага + OUT Y40 <u>1 шаг</u> <b>3 шага</b>





## 4 Структура разделов

Этот раздел является введением к разделам с 5-го по 9-й. В этом разделе разъяснена компоновка и структура пояснений к командам MELSEC серий "A"/"Q" и "System Q".

Как показано на следующей иллюстрации, каждый из вышеупомянутых разделов начинается с таблицы, в которой разъяснена структура обозначений команд, описываемых в данном разделе.

Вид команды	Значение
Операторы сравнения	Сравнение данных, например, =, >, ≥ и т. п.
Арифметические операции	Сложение, вычитание, умножение и деление двоичных и двоично-десятичных данных, чисел с плавающей запятой и двоичных блоков данных, соединение строковых величин, приращение, декремент
Команды конвертирования	Конвертирование данных, например, BCD → BIN или BIN → BCD
Команды переноса	Перенос, замена и инвертирование данных
Команды разветвления программы	Переходы, вызов подпрограммы
Команды для вызова программ прерывания	Вызов программ прерывания
Команды обновления данных	Обновление канала связи и интерфейса ввода-вывода
Прочие команды	Однофазные и двухфазные возрастающие и убывающие счетчики, программируемые таймеры, таймер особой функции, команда позиционирования, линейно изменяющийся сигнал, счетчик импульсов, импульсный выход, широтно-импульсная модуляция, матрица ввода

Каждая позиция рассматривается в отдельном подразделе и иллюстрируется примерами программ.

## 4.1 Обзор команд

Каждый подраздел начинается с таблицы, в которой перечислены все команды, разъясняемые в этом подразделе. Как показано на следующей иллюстрации, приводятся все варианты написания команды, как в редакторе MELSEC, так и в редакторе IEC.

### 6.1 Операторы сравнения

Операторы сравнения могут сравнивать два блока данных (например, равно =, больше >, меньше < и т. п.). Операторы сравнения программируются аналогично тому, как это описано для соответствующих команд из базового набора команд:

LD, LDI ⇒ LD=, LDD=

AND, ANI ⇒ AND=, ANDD=

OR, ORI ⇒ OR=, ORD=

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC	Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC
	LD=	LD_EQ_M		LD<=	LD_LE_M
	AND=	AND_EQ_M		AND<=	AND_LE_M
	OR=	OR_EQ_M		OR<=	OR_LE_M
	LDD=	LDD_EQ_M		LDD<=	LDD_LE_M
	ANDD=	ANDD_EQ_M		ANDD<=	ANDD_LE_M
	ORD=	ORD_EQ_M		ORD<=	ORD_LE_M
	LDE=	LD_EEQ_M		LDE<=	LD_ELE_M

Если в GX IEC Developer имеется возможность выбора, следует всегда использовать команду по стандарту Международной электротехнической комиссии (IEC).

## 4.2 Таблица процессоров

Подразделы, посвященные описанию отдельных команд, начинаются с таблицы. В ней указано, в каких центральных процессорах (AnS, AnN, AnA, AnAS, AnU, QnA, QnAS, Q4AR, Q) эта команда может быть выполнена. Процессоры, способные выполнить данную команду, помечены черным кружком.

**INT, INTP, DINT, DINTP** Команды конвертирования

---

### 6.3.4 INT, INTP, DINT, DINTP

<b>Процессор</b>	AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
			● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC - в сочетании с командами LEDA, C и R.

<sup>2</sup> Кроме процессоров Q00J, Q00 и Q01

Если в каком-либо процессоре выполнение команды имеет особенность, то в таблице имеется сноска на разъяснение этой особенности (например, "Специализированные команды" см. в п. 3.3 "Программирование специализированных команд").

## 4.3 Операнды MELSEC A

В таблице "Операнды MELSEC A" перечислены все имеющиеся операнды, которые можно использовать для внутренних переменных (например, s1, s2, d).

Операнды MELSEC A	Операнды																Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки					
	битовые							словные (16 бит)							Константы							Указатели		Уровень		
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						K	H (16#)	P	I	N
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	5/7 1	●		●	
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●									

<sup>1</sup> Если выполняется функция индексации, длина блока битового операнда не равна K4 или головной адрес битового операнда не кратен 8 (или 16 в случае процессоров A3H, A3M, AnA, AnAS или AnU), то расходятся 7 шагов. Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

Применимые битовые и словные операнды перечислены отдельно. Если операнд не применим, то он не обозначен черным кружком.

В столбце "Константы" указано, в каком виде команда может использовать константы – в десятичном (K) или шестнадцатеричном (H, 16#).

Столбец "Указатели" содержит информацию о том, может ли команда применять указатели (P) и/или указатели прерывания (I).

Если команда может выполняться во вложенных уровнях, это помечено в столбце "Уровень".

В столбце "Длина блока" указана длина блоков для битовых операндов, используемых командой. В вышеприведенном примере команда может адресовать блоки от K1 до K4, т. е. длиной от 4 до 16 битов.

В столбце "Шагов" указано количество используемых шагов программы.

Если команда может применять индексированную адресацию, это помечено в столбце "Индекс".

Столбец "Флаг переноса" показывает, может ли команда устанавливать флаг переноса при выполнении результата.

В столбце "Флаг ошибки" отмечается возможность установления флага ошибки.

Если имеются какие-либо особенности, то в таблице имеется сноска, а под таблицей – разъяснение.

## 4.4 Операнды MELSEC Q

Термин "MELSEC Q" охватывает все центральные процессоры MELSEC "System Q", а также центральные процессоры QnA, QnAS и Q4AR.

В таблице "Операнды MELSEC Q" перечислены все имеющиеся операнды, которые можно использовать для внутренних переменных (например, s1, s2, d).

Операнды не перечислены отдельно – указываются только типы операндов, которыми может оперировать команда (битовые и/или словные операнды).

Операнды MELSEC Q	Операнды										
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□G□	Индексные регистры Zn	Константы K, H (16#)	Иные	Флаг ошибки	Шагов
	битовые	словные		битовые	словные						
s1	●	●	●	●	●	●	●	●	—	—	3
s2	●	●	●	●	●	●	●	●	—	—	

Если команда имеет возможность обращаться к регистру файлов, это помечено в столбце "Регистр файлов".

В столбце "MELSECNET/10 непоср. адр. J□□" указывается, может ли команда записывать битовые и/или словные данные в станции, подключенные к сети MELSECNET/10, и считывать данные из этих станций. "J□□" обозначает номер станции, а "□□" – адрес операнда.

Столбец "Специальные модули U□G□" содержит информацию о том, может ли команда записывать данные в буферную память подключенного специального модуля и считывать данные из этой памяти. "U□□" указывает головной адрес специального модуля, а "G□□" – адрес в буферной памяти.

Если команда использует индексированную адресацию, это помечено в столбце "Индексный регистр Zn".

В столбце "Константы K, H (16#)" указано, в каком виде команда может использовать константы – в десятичном (K) или шестнадцатеричном (H, 16#).

В столбце "Прочее" указывается, использует ли команда какие-либо иные операнды и константы.

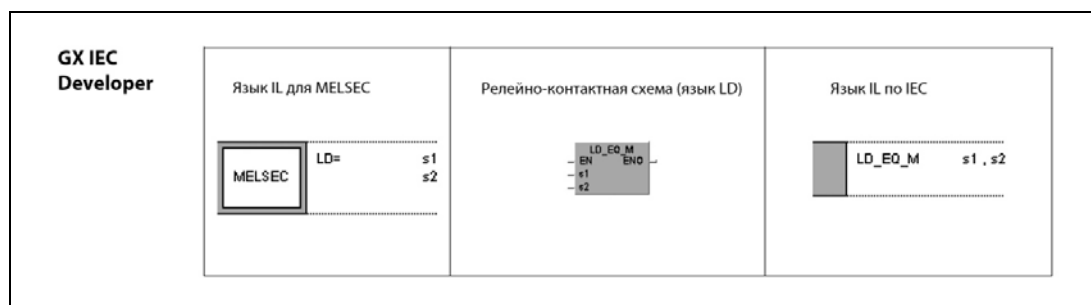
В столбце "Флаг ошибки" отмечается возможность установления флага ошибки.

В столбце "Шагов" указано количество расходуемых шагов программы.

### 4.4.1 Отображение в GX IEC Developer

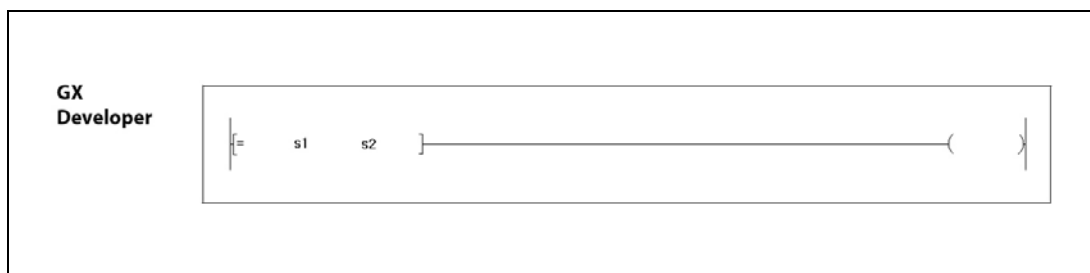
Вслед за таблицами операндов указываются форматы отображения команды в GX IEC Developer.

На рисунке ниже, слева направо, команда LD\_EQ\_M отображена в редакторе MELSEC (на языке IL для контроллеров MELSEC) и в редакторе IEC (на языке релейно-контактных схем и языке IL по стандарту МЭК).



## 4.4.2 Отображение в GX Developer

За форматами команды в GX IEC Developer следуют форматы в GX Developer.



## 4.5 Переменные

Таблица переменных содержит все внутренние переменные команды.

Переменные	Операнд	Значение	Тип данных	
			MELSEC	IEC
s	s+0:	измерение частоты вращения (только для внутрисистемного использования)	BIN 16 битов	массив [1..3] данных типа ANY16
	s+1:	номер позиции		
	s+2:	номер сектора		
n1		количество делений (от 2 до 32767)		ANY16
n2		количество делений для низкой скорости вращения (ползучей скорости) (от 0 до n1).		ANY16
d	d+0:	входной сигнал (фаза "A")	бит	массив [1..8] переменных типа Bool
	d+1:	входной сигнал (фаза "B")		
	d+2:	входной сигнал определения нулевой точки		
	d+3:	выходной сигнал "вперед, высокая частота вращения" (внутрисистемный)		
	d+4:	выходной сигнал "вперед, низкая частота вращения" (внутрисистемный)		
	d+5:	выходной сигнал "стоп" (внутрисистемный)		
	d+6:	выходной сигнал "назад, высокая частота вращения" (внутрисистемный)		
d+7:	выходной сигнал "назад, низкая частота вращения" (внутрисистемный)			

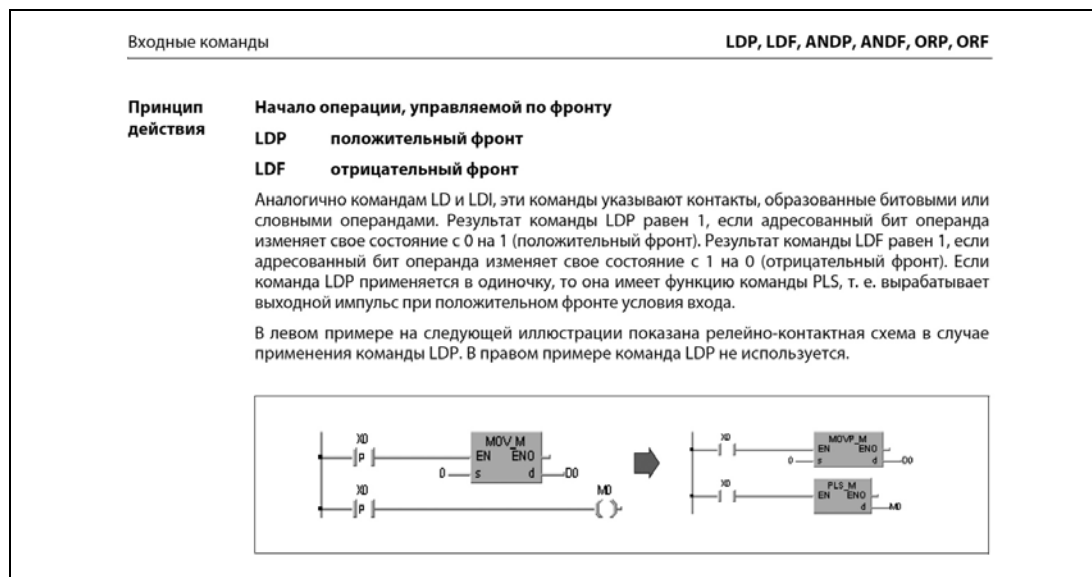
В столбце "Значение" разъяснена функция операндов и элементов операндов.

В столбце "Тип данных" перечислены типы данных операндов. Если в отношении типов данных имеются различия между редакторами MELSEC и IEC, эти различия также разъясняются. Дополнительную информацию на тему "Переменные" вы найдете в разделах 3.4 "Программирование переменных" и 3.5 "Типы данных".

## 4.6 Принцип действия

Пункт "Принцип действия" подробно разъясняет принцип действия команды.

На рисунке ниже показано описание принципа действия команды LDF/LDP.



## 4.7 Примечания

В примечаниях указывается на особенности, возможные ошибки и риски при программировании команд.

**Примечания** При некоторых обстоятельствах, если к предшествующим контактам обращается подпрограмма или цикл FOR/NEXT, команды MEP и MEF могут выполняться неправильно. В этом случае следует применять команду EGP/EGF.

В связи с тем, что команды MEP/MEF работают с сигналами, непосредственно предшествующими командам MEP/MEF, перед этими командами следует программировать команду AND. Команды MEP/MEF невозможно программировать в сочетании с командой LD или OR.

## 4.8 Источники ошибок

Описание источников ошибок относится, преимущественно, к кодам ошибок серий "Q" и "System Q" (см. 11.1 "Перечень кодов ошибок (серии Q00J, Q00 и Q01)" и 11.2 "Перечень кодов ошибок (серии QnA и System Q)"). Коды ошибок серии "A" перечислены в разделах 11.3 "Перечень кодов ошибок серии A (кроме AnA и AnAS)" и 11.4 "Перечень кодов ошибок серий AnA и AnAS".

На рисунке ниже показаны источники ошибок команды DELTA/DELTAP.

**Источники ошибок** В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанный в d выход находится вне диапазона адресов для выходов (код ошибки 4101).

## 4.9 Примеры

В качестве примеров программ приводятся, прежде всего, программы для процессоров серии "Q". Примеры программ представлены на языке IL для MELSEC, а также на языке релейно-контактных схем (LD) и языке IL по стандарту МЭК. Для лучшего понимания во многих случаях добавлены графики.

На рисунке ниже показан пример программы для команд LD, AND, OR и ORI.

**Пример 1** LD, AND, OR, ORI

В следующей программе показаны последовательные и параллельные схемы контактов. Бит 5 (b5) в D0 также опрашивается в качестве контакта.

Язык IL для MELSEC		
MELSEC	LD	X3
	OR	D0.5
	OUT	Y33
MELSEC	LD	X5
	AND	M11
	OUT	Y34

Язык IL по IEC		
LD	X3	
OR	D0.5	
ST	Y33	
LD	X5	
AND	M11	
ST	Y34	

b15 ----- b5 ----- b0

D0

На рисунке ниже показан пример программы для команды RBMOV, написанной в среде GX Developer.

**Пример 1** RBMOV

Следующая программа при положительном фронте сигнала SM402 передает данные 4 младших битов (от b0 до b3) из регистров D66...D69 на выходы Y30...Y3F. Количество передаваемых блоков (4) указано константой K4.

На битовых схемах показана структура битов перед передачей и после передачи.

Список инструкций				
0	LD	SM402		
1	RBMOV	D66	K1Y30	K4
5	END			

	b15	b14	b3	b0																						
D66		1	1	0	1	<table border="1"> <tr> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>Y33 - Y30</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>Y37 - Y34</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>Y3B - Y38</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>1</td> <td>Y3F - Y3C</td> </tr> </table>	1	1	0	1	Y33 - Y30	0	0	0	0	Y37 - Y34	0	0	1	1	Y3B - Y38	1	1	0	1	Y3F - Y3C
1	1	0	1	Y33 - Y30																						
0	0	0	0	Y37 - Y34																						
0	0	1	1	Y3B - Y38																						
1	1	0	1	Y3F - Y3C																						
D67		0	0	0	0																					
D68		1	0	0	1																					
D69		0	1	1	0	1																				

<sup>1</sup> эти биты при операции игнорируются





## 5

# Базовый набор команд


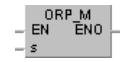
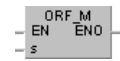


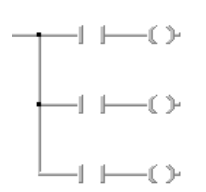
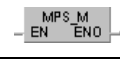
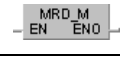
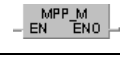
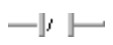
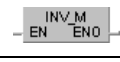


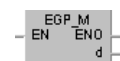

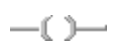


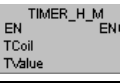
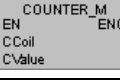
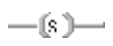
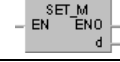
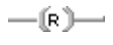
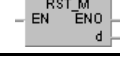
Помимо обычных команд для программирования входных и выходных контактов, базовый набор команд содержит команды перехода, команды соединения блоков, команды сдвига битов, команды организации и конца программы, а также прочие команды, образующие основной каркас для программирования контроллеров серии MELSEC.

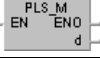
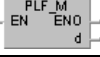
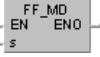
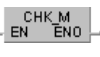
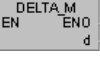
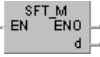
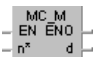
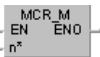
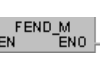
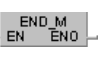

Обзор базового набора команд дан в следующих таблицах.

Разновидность	Пояснение
Входные команды	Начало операции, параллельное и последовательное соединение контактов
Соединяющие команды	Последовательное и параллельное соединение блоков, сохранение и обработка результата логической операции, инвертирование сигнала результата операции, преобразование результата операции в импульсный вид, установка маркеров фронта
Выходные команды	Битовые операнды, контакты счетчиков и таймеров, вывод, установка и сброс маркеров ошибки, установка и сброс операндов, вывод с управлением по фронту сигнала, инвертирование коммутационного состояния операнда, выработка переключающих импульсов
Команды сдвига	Сдвиг битовых операндов
Организирующие команды	Активация и деактивация отдельных областей программы
Команды конца программы	Конец области программы, конец главных программ и подпрограмм
Прочие команды	Прерывание обработки, пустой шаг в программе

**ПРИМЕЧАНИЕ** В следующей таблице помимо команд MELSEC в различных редакторах перечислены соответствующие команды по стандарту МЭК (IEC).

Команда MELSEC				Команда IEC в редакторе IEC
в редакторе MELSEC	в редакторе IEC			
	Язык IL	Язык LD		
LD	—		—	LD
LDI	—		—	LDN
AND	—			AND
ANI	—			ANDN
OR	—		—	OR
ORI	—		—	ORN
LDP	LDP_M	—	—	—
LDF	LDF_M	—	—	—
ANDP	ANDP_M	—		—

Команда MELSEC				Команда IEC в редакторе IEC
в редакторе MELSEC	в редакторе IEC			
	Язык IL	Язык LD		
ANDF	ANDF_M	—		—
ORP	ORP_M	—		—
ORF	ORF_M	—		—
ANB	—		—	AND ( ... )
ORB	—		—	OR ( ... )
MPS	MPS_M			—
MRD	MRD_M			—
MPP	MPP_M			—
INV	INV_M			NOT
MEP	MEP_M	—		—
MEF	MEF_M	—		—
EGP	EGP_M	—		—
EGF	EGF_M	—		—
OUT	OUT_M			ST
OUT (T)	TIMER_M	—		—
OUT H (T)	TIMER_H_M	—		—
OUT (C)	COUNTER_M	—		—
SET	SET_M			S
RST	RST_M			R

Команда MELSEC				Команда IEC в редакторе IEC
в редакторе MELSEC	в редакторе IEC		Язык LD	
	Язык IL			
PLS	PLS_M	—		R_TRIG ● <sup>1</sup>
PLF	PLF_M	—		F_TRIG ● <sup>1</sup>
FF	FF_M	—		—
CHK	CHK_M	—		—
DELTA	DELTA_M	—		—
SFT	SFT_M	—		SHL/SHR
MC	MC_M	—		—
MCR	MCR_M	—		—
FEND	FEND_M	—		● <sup>2</sup>
END	END_M	—		● <sup>2</sup>
STOP	STOP_M	—		—
NOP	—	—	—	—

<sup>1</sup> Подразумеваются функциональные блоки IEC.

<sup>2</sup> Команды FEND и END автоматически вырабатывает среда программирования GX Developer и GX IEC Developer.

## 5.1 Входные команды

### 5.1.1 LD, LDI, AND, ANI, OR, ORI

#### Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

#### Операнды MELSEC A

Операнды														Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9011							
Битовые операнды				Словные операнды (16 бит)						Константы		Указатели							Уровень						
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1						Z	V	K	N (16#)	P	I	N
●	●	●	●	●	●	●	●	●														1	1		

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

#### Операнды MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. Адр. J□□		Спец. модули U□□G□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные				DX, BL			
s	●	●	●	●	●	—	—	●	—	—	1

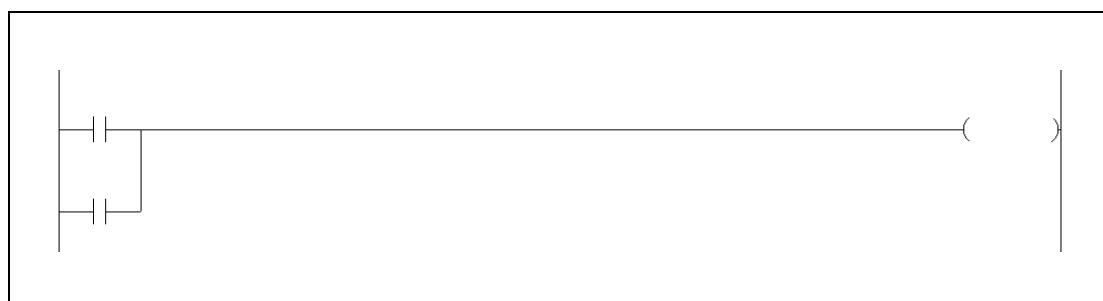
<sup>1</sup> Количество шагов может быть различным:

- при использовании внутреннего операнда или регистров файлов с R0 по R32767: 1 шаг
- при использовании непосредственно адресуемых входов (DX): 2 шага
- при использовании иных операндов: 3 шага

#### GX IEC Developer



#### GX Developer



#### Переменные

Операнд	Значение	Тип данных
s	Операнды устанавливаются в качестве контактов.	бит

**Принцип действия****Начало операции****LD**      **Загрузить (закрывающий контакт)****LDI**     **Загрузить инверсно (размыкающий контакт)**

Операция всегда начинается с команды LD (**LoaD**; "загрузить") или LDI (**LoaD Inverse**; загрузить инверсно). Команда LD указывает закрывающий контакт, а команда LDI – размыкающий контакт. Содержащийся в команде адрес операнда является входным условием для следующей команды.

**Последовательное соединение****AND**     **закрывающих контактов****ANI**     **размыкающих контактов**

Для последовательного соединения контактов используется команда AND в качестве закрывающего контакта или команда ANI в качестве размыкающего контакта. Содержащийся в команде операнд является условием дальнейшего переключения к следующей команде.

Обе команды представляют собой логические связи и их нельзя программировать в начале операции.

**Параллельное соединение****OR**      **закрывающих контактов****ORI**     **размыкающих контактов**

Для параллельного соединения контактов используется команда OR в качестве закрывающего контакта или команда ORI в качестве размыкающего контакта. Содержащийся в командах операнд является условием дальнейшего переключения от предшествующей команды к последующей.

Обе команды представляют собой логические связи и их нельзя программировать в начале операции.

**ПРИМЕЧАНИЯ**

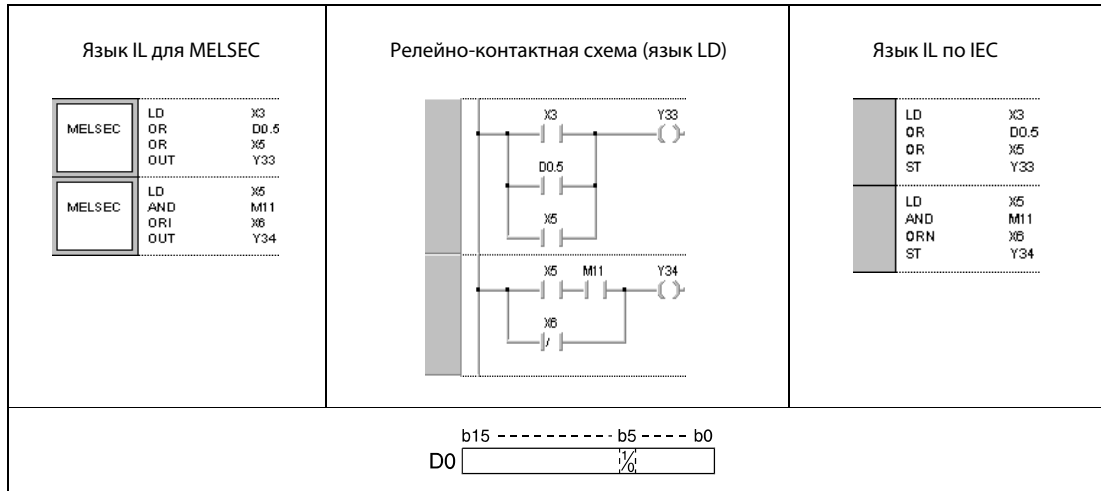
*Указанные в командах операнды могут быть словными. В этом случае в качестве состояния контакта считается состояние выбранного бита (только в серии "Q" и "System Q").*

*Опрашиваемый бит словного операнда указывается в шестнадцатеричном виде. Например, бит b11 в D0 указывается в виде D0.0B (только в серии "Q" и "System Q").*

*Дополнительная информация об адресации битов в словных операндах имеется в разделе "Состав команд" руководства (только в серии "Q" и "System Q").*

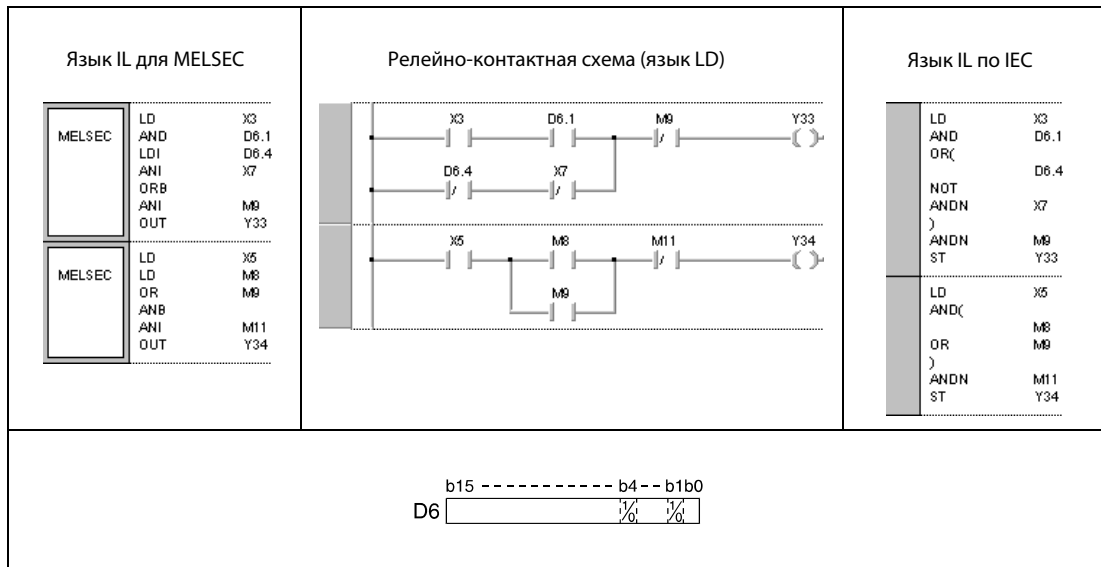
**Пример 1** LD, AND, OR, ORI

В следующей программе показаны последовательные и параллельные схемы контактов. Бит 5 (b5) в D0 также опрашивается в качестве контакта.



**Пример 2** LD, LDI, AND, ANI, OR

В следующей программе показаны смешанные схемы. Некоторые контактные точки схемы связаны командами ORB и ANB. Адресованные биты (b1 и b4) из регистра D6 считываются в качестве контактов.



**Пример 3** LD, AND, ANI

В следующей программе результаты команд выводятся в операнды с Y35 по Y37.



## 5.1.2 LDP, LDF, ANDP, ANDF, ORP, ORF

## Процессор

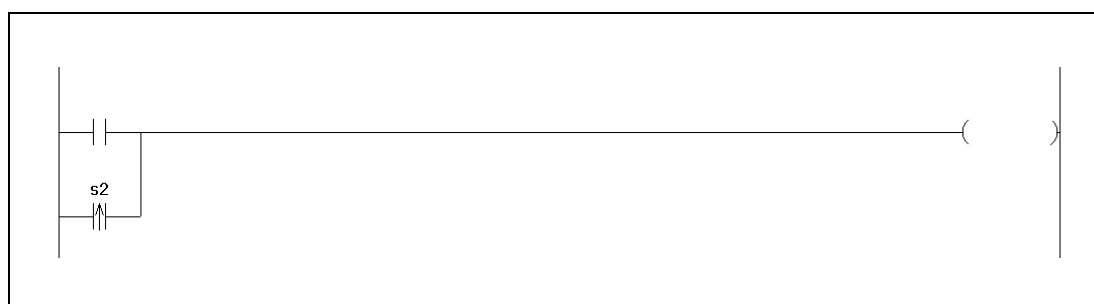
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Операнды  
MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные DX		
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	—	—	●	—	2 <sup>1</sup>

<sup>1</sup> Количество шагов может быть различным:

- при использовании внутреннего операнда или регистров файлов с R0 по R32767: 2 шага
- при использовании непосредственно адресуемых входов (DX): 3 шага
- при использовании иных операндов: 4 шага

GX IEC  
DeveloperGX  
Developer

## Переменные

Операнд	Значение	Тип данных
s	Операнды устанавливаются в качестве контактов.	бит



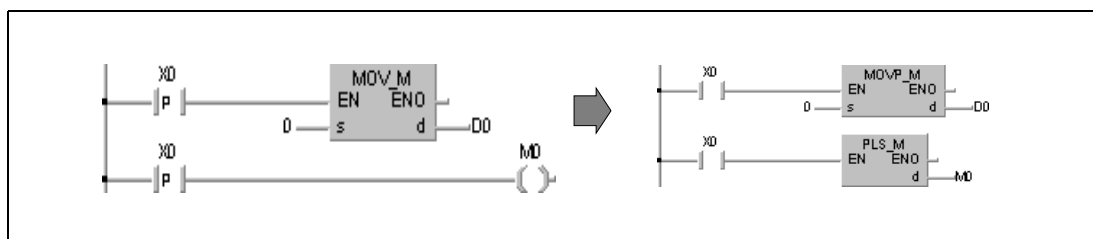
**Принцип действия** Начало операции, управляемой по фронту

**LDP**      **положительный фронт**

**LDF**      **отрицательный фронт**

Аналогично командам LD и LDI, эти команды указывают контакты, образованные битовыми или словными операндами. Результат команды LDP равен 1, если адресованный бит операнда изменяет свое состояние с 0 на 1 (положительный фронт). Результат команды LDF равен 1, если адресованный бит операнда изменяет свое состояние с 1 на 0 (отрицательный фронт). Если команда LDP применяется в одиночку, то она имеет функцию команды PLS, т. е. вырабатывает выходной импульс при положительном фронте условия входа.

В левом примере на следующей иллюстрации показана релейно-контактная схема в случае применения команды LDP. В правом примере команда LDP не используется.



**Последовательное соединение с управлением по фронту**

**ANDP**      **положительный фронт**

**ANDF**      **отрицательный фронт**

Команда ANDP последовательно соединяет контакт с контактом, образованным битовым или словным операндом. Этот контакт имеет состояние 1, если адресованный бит операнда изменяет состояние с 0 на 1. В случае команды ANDF образованный контакт имеет состояние 1, если адресованный бит операнда изменяет состояние с 1 на 0.

**Параллельное соединение с управлением по фронту**

**ORP**      **положительный фронт**

**ORF**      **отрицательный фронт**

Команда ORP параллельно соединяет контакт с контактом, образованным битовым или словным операндом. Образованный контакт имеет состояние 1, если адресованный бит операнда изменяет состояние с 0 на 1. В случае команды ORF образованный контакт имеет состояние 1, если адресованный бит операнда изменяет состояние с 1 на 0.

Операнд, указанный в команде ANDP/ORP	Результат команды ANDP/ORP	Операнд, указанный в команде ANDF/ORF	Результат команды ANDF/ORF
Битовый/словный операнд		Битовый/словный операнд	
0 → 1	1	0 → 1	0
0	0	0	
1		1	
1 → 0		1 → 0	1

**ПРИМЕЧАНИЕ** *Опрашиваемые биты в словных операндах адресуются в шестнадцатеричном виде. Например, бит 11 в D0 адресуются в виде D0.0B.*

**Пример** ORP

В следующей программе при положительном фронте сигнала X0 или при установке (положительном фронте) бита 10 (b10) в регистре данных D0 выполняется команда MOV.



## 5.2 Соединяющие команды

### 5.2.1 ANB, ORB

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

Операнды																Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки				
Битовые операнды						Словные операнды (16 бит)						Константы	Указатели	Уровень	M9012				M9010 M9011					
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V	K	H (16#)	P	I
																						1		

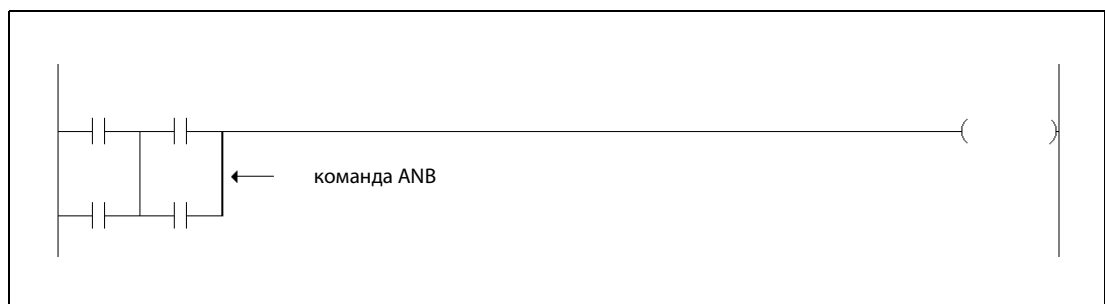
Операнды MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 неоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные							
—	—	—	—	—	—	—	—	—	—	—	1

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
—	—	—

**Принцип действия**

**Последовательное соединение параллельных схем**

**ANB Последовательное соединение блоков**

Команда ANB ("И" для блоков) последовательно соединяет 2 или более параллельных схемы и образует условие для дальнейшего переключения. Если последовательно соединяются несколько блоков, то команду ANB необходимо запрограммировать после каждого из параллельных блоков.

Соединение ANB является независимой командой и не нуждается в операндах.

В одной и той же программе команду ANB можно использовать сколь угодно часто.

Если соединяются несколько блоков непосредственно один за другим, то максимальное количество команд ANB для процессоров QnA, AnA, AnAS и AnU равно 15 (т. е., соответственно, максимальное число блоков равно 16), а для всех прочих центральных процессоров – 7 (соответственно, 8 блоков). При превышении соответствующих пределов безупречная работа более не обеспечивается.

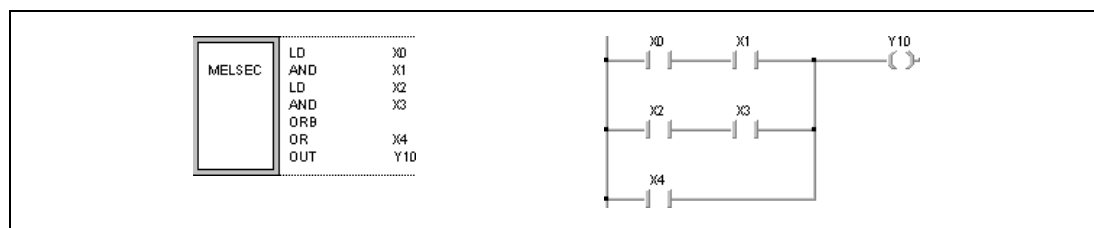
**Параллельное соединение последовательных схем**

**ORB Параллельное соединение блоков**

Команда ORB ("ИЛИ" для блоков) параллельно соединяет 2 или более последовательных схем и образует условие дальнейшего прохождения сигнала.

Если параллельно соединяются несколько блоков, то команду ORB необходимо запрограммировать после каждого из этих блоков.

При параллельном соединении блоков, в которых имеется лишь один контакт, вместо команды ORB следует использовать команду OR или ORI.



Соединение ORB является независимой командой, не нуждающейся ни в одном операнде.

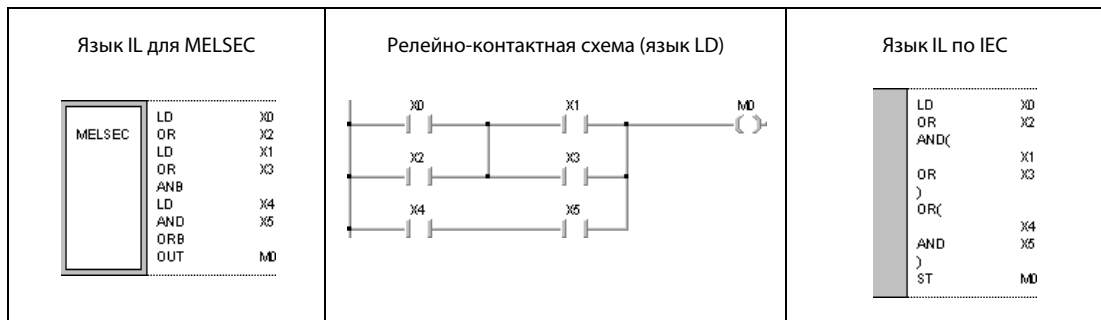
В одной и той же программе команду ORB можно использовать сколь угодно часто.

Если соединяются несколько блоков, находящихся непосредственно рядом друг с другом, то максимальное количество команд ORB для процессоров QnA, AnA, AnAS и AnU равно 15 (соответственно, максимальное число блоков равно 16), а для всех прочих центральных процессоров – 7 (соответственно, 8 блоков). При превышении соответствующих пределов безупречная работа более не обеспечивается.

**Пример**

ANB, ORB

В следующей программе параллельное соединение X0 и X2 последовательно соединяется с параллельным соединением X1 и X3. Результат параллельно соединяется с последовательным соединением X4 и X5.



### 5.2.2 MPS, MRD, MPP

**ПРИМЕЧАНИЕ** В редакторах по стандарту IEC эти команды использовать не следует.

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

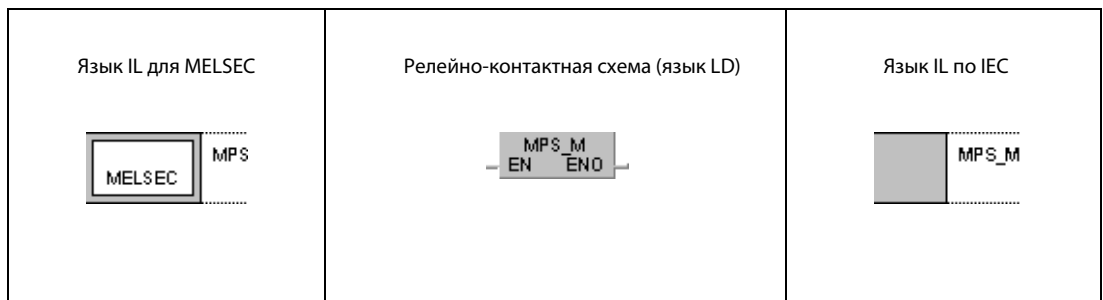
**Операнды MELSEC A**

Операнды																	Длина блока	Шагов	Индекс	Флаг переноса					
Битовые операнды							Словные операнды (16 бит)							Константы		Указатели				Уровень		M9012	M9010 M9011		
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K				N (16#)	P			I	N
																							1		

**Операнды MELSEC Q**

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zп	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные							
—	—	—	—	—	—	—	—	—	—	—	1

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
—	—	—

**Принцип действия****Обработка результата****MPS Сохранение результата**

Команда MPS сохраняет результат соединения, расположенного перед командой MPS.

Для процессоров Q, QnA, AnA, AnAS и AnU можно запрограммировать до 16 команд MPS подряд на каждую сеть, а для всех прочих центральных процессоров – до 12 команд. Если между запрограммированными подряд командами MPS имеется команда MPP, максимальное количество уменьшается на 1.

**MRD Считывание результата**

Команда MRD считывает результат соединения, сохраненный с помощью команды MPS. Следующий шаг выполняется в зависимости от считанного результата.

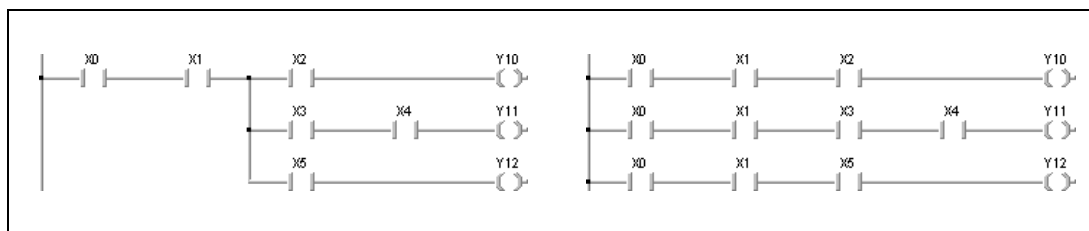
**MPP Считывание и стирание результата**

Команда MPP считывает результат соединения, сохраненный с помощью команды MPS, выполняет следующий шаг в зависимости от считанного результата и стирает результат.

Команды MPS, MRP или MPP являются независимыми командами и не нуждаются в операндах.

На языке релейно-контактных схем команды MPS, MRD и MPP отдельно не отображаются. Являются ли соединения соединениями типа MPS, MRD или MPP, зависит от конфигурации соединений релейно-контактной схемы.

На левой иллюстрации показана релейно-контактная схема с командами MPS и MRD. Правая релейно-контактная схема запрограммирована без применения команд MPS, MRD и MPP.



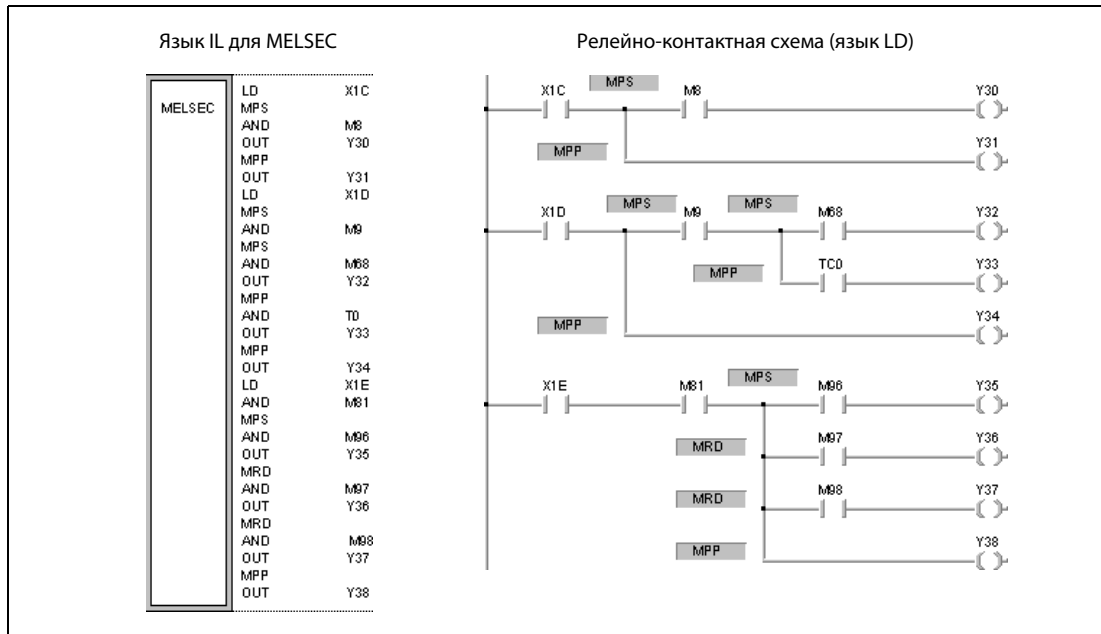
Количество команд MPS в программе должно совпадать с количеством команд MPP.

Если количество команд MPS превышает количество команд MPP, то вместо команды MPP устанавливается команда NOP и работа программы соответственно изменяется.

Если количество команд MPP превышает количество команд MPS, логическая череда команд программы прерывается. Выполнение программы останавливается и программируемый контроллер выдает сообщение об ошибке.

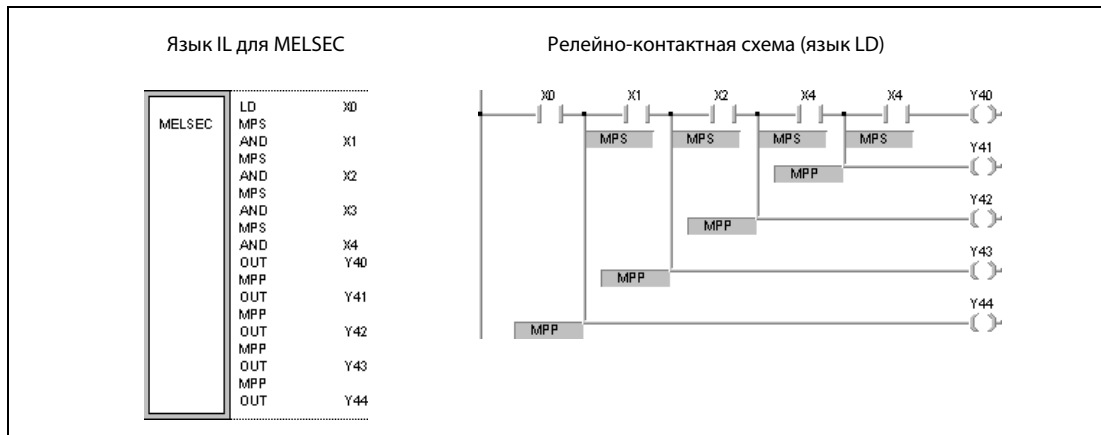
**Пример 1** MPS, MRD, MPP

Следующий пример программы поясняет применение команд при программировании смешанных схем.



**Пример 2** MPS, MRD, MPP

Следующий пример программы поясняет программирование команд для вывода промежуточных результатов последовательного соединения.





### 5.2.3 INV

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

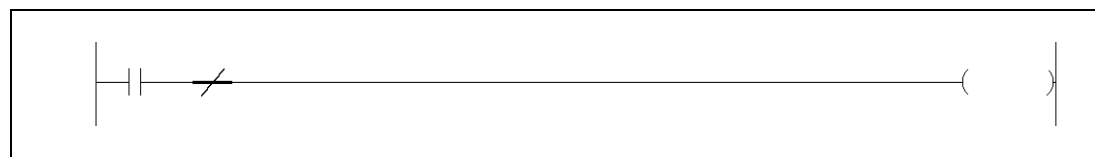
**Операнды MELSEC Q**

Операнды								Флаг ошибки	Шагов	
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные U
битовые	словные		битовые	словные						
—	—	—	—	—	—	—	—	—	1	

**GX IEC Developer**

<p>Язык IL для MELSEC</p> 	<p>Релейно-контактная схема (язык LD)</p> 	<p>Язык IL по IEC (команда по IEC)</p> 
---	--	--

**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
—	—	—

**Принцип действия**

**Инвертирование сигнала результата операции**

**INV Команда инвертирования**

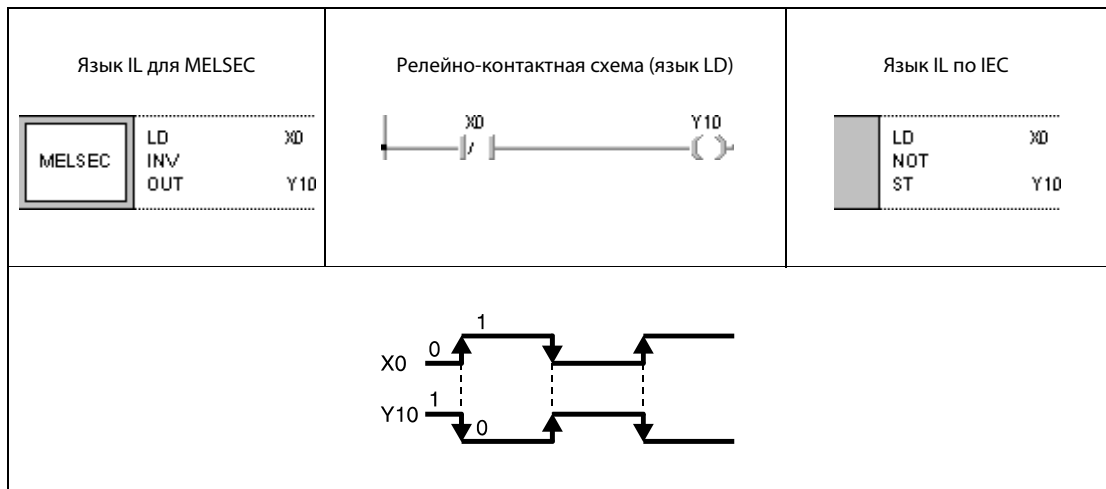
Команда INV изменяет результат операции, расположенной перед этой командой, на противоположный.

Если перед выполнением команды результат был равен 1, то после выполнения он становится равным 0.

Если перед выполнением команды результат был равен 0, то после выполнения он становится равным 1.

**Пример**

В следующей программе результат операции X0 изменяется на противоположный, и инвертированный сигнал выводится на Y10.



### 5.2.4 MER, MEF

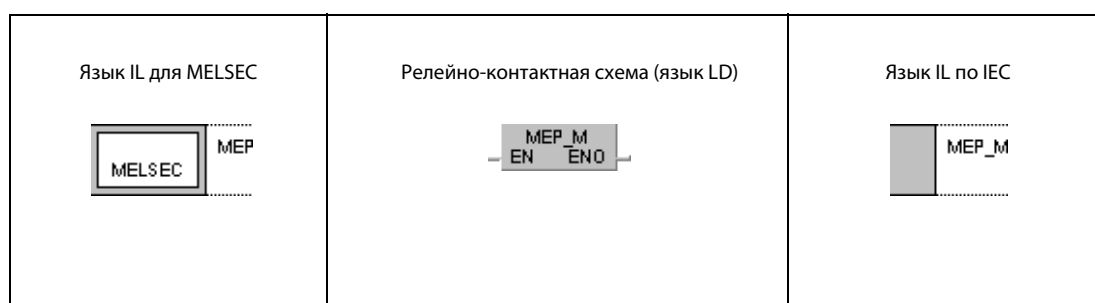
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

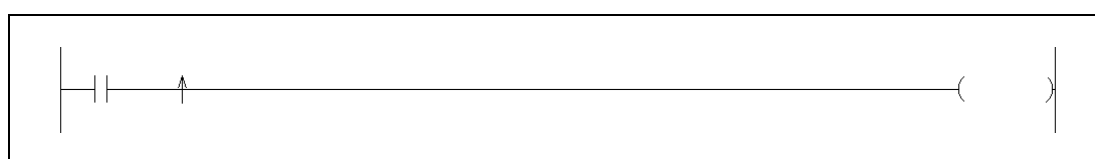
**Операнды MELSEC Q**

Операнды								Флаг ошибки	Шагов	
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные U
битовые	словные		битовые	словные						
—	—	—	—	—	—	—	—	—	1	

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
—	—	—

**Принцип действия****Преобразование результатов операций в импульсный вид****MEP Выработка импульса при положительном фронте результата операции**

Команда MEP применяется в тех случаях, если используемые команды не способны выдавать результат операции в виде определенного выходного импульса. В таких случаях за соответствующей командой программируется команда MEP, которая при изменении входного сигнала с 0 на 1 (положительном фронте сигнала) вырабатывает единственный выходной импульс. Следующий импульс вырабатывается лишь при очередном нарастающем фронте импульса на входе этой команды.

**MEF Выработка импульса при отрицательном фронте результата операции**

Команда MEF работает аналогично команде MEP, с той разницей, что здесь одноразовый выходной импульс вырабатывается при изменении входного сигнала с 1 на 0 (т. е. при отрицательном фронте). Следующий импульс вырабатывается лишь при очередном понижающемся фронте на входе.

Обе эти команды особенно полезны при использовании нескольких соединенных друг с другом контактов. Например, несколько последовательно соединенных замыкающих контактов в замкнутом состоянии в качестве результата операции постоянно вырабатывали бы 1. Если на основе этого результата устанавливается маркер, то сброс этого маркера был бы невозможен. Путем последовательного соединения с командой MEP сброс становится возможным, так как устанавливается только один выходной импульс – в тот момент, когда выходное состояние последовательного соединения меняется с 0 на 1.

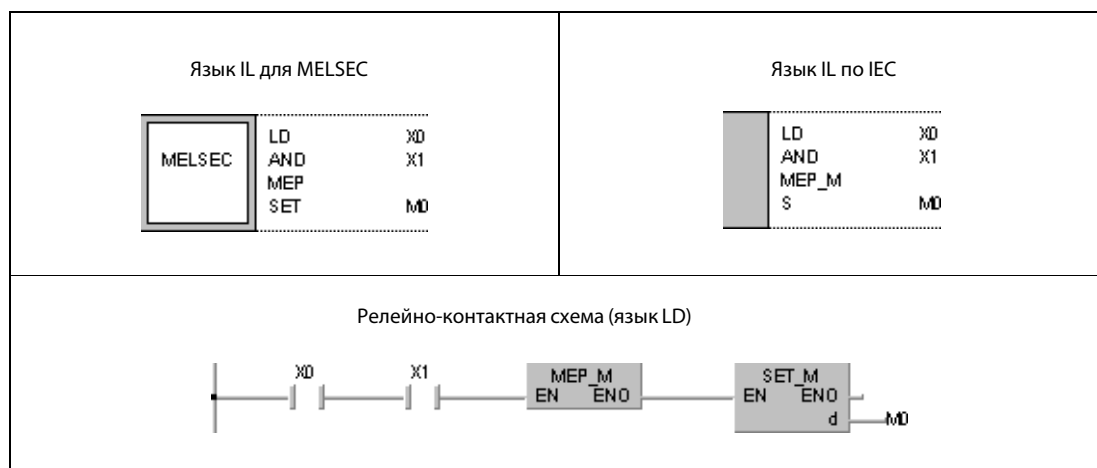
**ПРИМЕЧАНИЯ**

При некоторых обстоятельствах, если к предшествующим контактам обращается подпрограмма или цикл FOR/NEXT, команды MEP и MEF могут выполняться неправильно. В этом случае следует применять команду EGP/EGF.

В связи с тем, что команды MEP/MEF работают с сигналами, непосредственно предшествующими командам MEP/MEF, перед этими командами следует программировать команду AND. Команды MEP/MEF невозможно программировать в сочетании с командой LD или OR.

**Пример****MEP**

В следующем примере программы при положительном фронте результата последовательного соединения X0 и X1 на основе выходного импульса команды MEP устанавливается маркер M0.



## 5.2.5 EGP, EGF

## Процессор

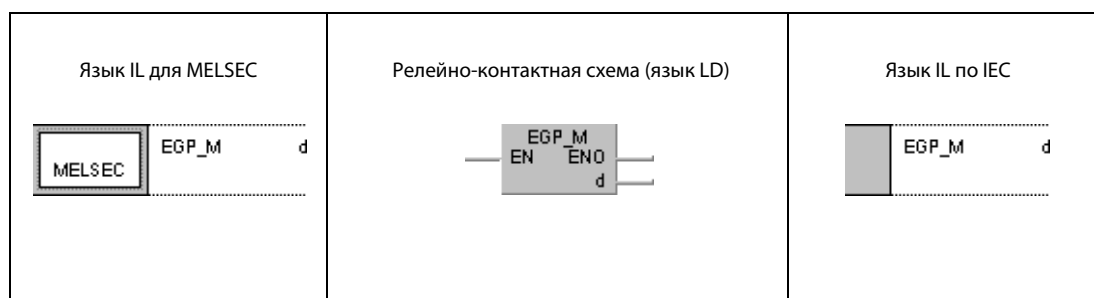
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

## Операнды MELSEC Q

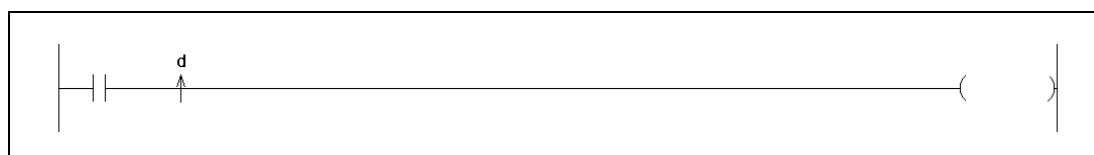
	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные U
	битовые	словные		битовые	словные						
d	● <sup>1</sup>	—	—	—	—	—	—	—	—	1	

<sup>1</sup> только V

## GX IEC Developer



## GX Developer



## Переменные

Операнд	Значение	Тип данных
d	Маркер фронта, в котором сохраняется результат.	бит (только V)

## Принцип действия

## Установка маркера фронта

**EGP Установка маркера фронта при положительном фронте результата операции**

Команда EGP устанавливает маркер фронта (V) в зависимости от результата операции предшествующей команды. Если результат изменяется с 0 на 1, маркер фронта устанавливается. При всех других входных состояниях команды EGP (например, при изменении состояния с 1 на 0 или удержании состояний 1 или 0) маркер фронта не устанавливается.

**EGF Установка маркера фронта при отрицательном фронте результата операции**

Команда EGF действует аналогично команде EGP, однако маркер фронта (V) устанавливается, если результат предшествующей команды изменяется с 1 на 0. При изменении результата с 0 на 1 или удержании состояний 1 или 0 маркер фронта не устанавливается.

Команды EGP/EGF используются в подпрограммах или программах, использующих адресацию с помощью индексных регистров (индексную адресацию) и находящихся между командами FOR и NEXT.

Команду EGP/EGF можно использовать как команду AND.

**Пример** EGP

В следующем примере сначала индексный регистр Z0 устанавливается на 0, а затем вызывается подпрограмма UP1 (1). В ней при положительном фронте X0Z0 устанавливается на X0, а V0Z0 на V0. Кроме того, D0Z0 устанавливается на D0 и получает приращение на 1.

После возврата в основную программу в индексный регистр Z0 записывается 1 и снова вызывается подпрограмма (2). При положительном фронте X1 устанавливается V1 и получает приращение D1.

Язык IL для MELSEC	Язык IL по IEC																																																																																															
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; text-align: center;">MELSEC</td> <td style="width: 15%;">LD</td> <td style="width: 15%;">SM400</td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> </tr> <tr> <td></td> <td>MOV</td> <td>K0</td> <td>Z0</td> <td></td> </tr> <tr> <td></td> <td>CALL</td> <td>UP1</td> <td></td> <td></td> </tr> <tr> <td></td> <td>MOV</td> <td>K1</td> <td>Z0</td> <td></td> </tr> <tr> <td></td> <td>CALL</td> <td>UP1</td> <td></td> <td></td> </tr> <tr> <td></td> <td>FEND</td> <td></td> <td></td> <td></td> </tr> <tr> <td style="text-align: center;">UP1:</td> <td>LD</td> <td>X0Z0</td> <td></td> <td></td> </tr> <tr> <td style="text-align: center;">MELSEC</td> <td>EGP</td> <td>V0Z0</td> <td></td> <td></td> </tr> <tr> <td></td> <td>INC</td> <td>D0Z0</td> <td></td> <td></td> </tr> <tr> <td></td> <td>RET</td> <td></td> <td></td> <td></td> </tr> </table>	MELSEC	LD	SM400				MOV	K0	Z0			CALL	UP1				MOV	K1	Z0			CALL	UP1				FEND				UP1:	LD	X0Z0			MELSEC	EGP	V0Z0				INC	D0Z0				RET				<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;"></td> <td style="width: 15%;">LD</td> <td style="width: 15%;">SM400</td> <td style="width: 15%;"></td> <td style="width: 15%;"></td> </tr> <tr> <td></td> <td>MOVE_E</td> <td>0, Z0</td> <td></td> <td></td> </tr> <tr> <td></td> <td>JMP</td> <td>UP1</td> <td></td> <td></td> </tr> <tr> <td></td> <td>LD</td> <td>SM400</td> <td></td> <td></td> </tr> <tr> <td></td> <td>MOVE_E</td> <td>1, Z0</td> <td></td> <td></td> </tr> <tr> <td></td> <td>JMP</td> <td>UP1</td> <td></td> <td></td> </tr> <tr> <td style="text-align: center;">UP1:</td> <td>LD</td> <td>X0Z0</td> <td></td> <td></td> </tr> <tr> <td></td> <td>EGP_M</td> <td>V0Z0</td> <td></td> <td></td> </tr> <tr> <td></td> <td>INC_M</td> <td>D0Z0</td> <td></td> <td></td> </tr> </table>		LD	SM400				MOVE_E	0, Z0				JMP	UP1				LD	SM400				MOVE_E	1, Z0				JMP	UP1			UP1:	LD	X0Z0				EGP_M	V0Z0				INC_M	D0Z0		
MELSEC	LD	SM400																																																																																														
	MOV	K0	Z0																																																																																													
	CALL	UP1																																																																																														
	MOV	K1	Z0																																																																																													
	CALL	UP1																																																																																														
	FEND																																																																																															
UP1:	LD	X0Z0																																																																																														
MELSEC	EGP	V0Z0																																																																																														
	INC	D0Z0																																																																																														
	RET																																																																																															
	LD	SM400																																																																																														
	MOVE_E	0, Z0																																																																																														
	JMP	UP1																																																																																														
	LD	SM400																																																																																														
	MOVE_E	1, Z0																																																																																														
	JMP	UP1																																																																																														
UP1:	LD	X0Z0																																																																																														
	EGP_M	V0Z0																																																																																														
	INC_M	D0Z0																																																																																														

Релейно-контактная схема (язык LD)

## 5.3 Команды для выходных контактов

### 5.3.1 OUT

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

d	Операнды																Длина блока	Шагов	Индекс	Флаг переноса		Флаг ошибки					
	Битовые операнды						Словные операнды (16 бит)						Константы		Указатели					Уровень							
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V				K	H (16#)		P	I	N	M9012	M9010 M9011
	●	●	●	●	●																			● <sup>1</sup>	● <sup>2</sup>		

<sup>1</sup> В принципе 1 шаг, однако если в качестве операндов команды OUT используются специальные маркеры или маркеры ошибки, то расходуются 3 шага. Количество шагов при использовании процессоров AnA, AnAS или AnU можно узнать в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

<sup>2</sup> Функция индекса имеется только в случае процессоров AnA, AnAS или AnU.

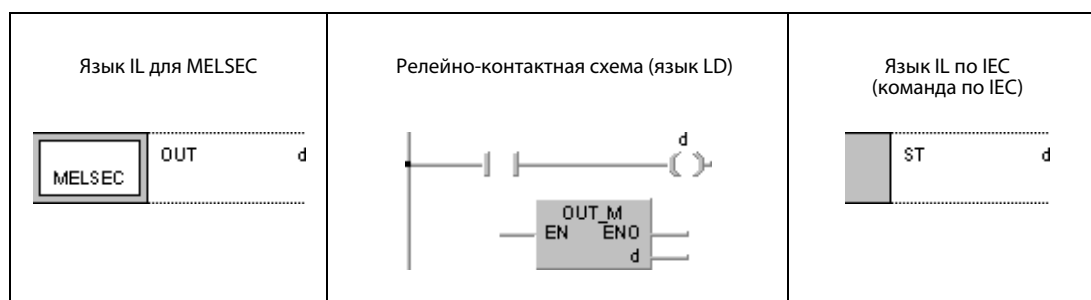
Операнды MELSEC Q

d	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
	● <sup>1</sup>	●	●	●	●	●	—	—	●	—	● <sup>2</sup>

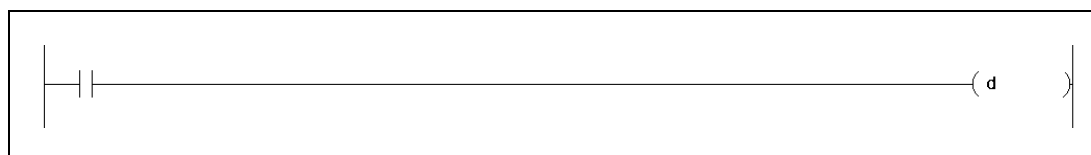
<sup>1</sup> кроме T, C, F

<sup>2</sup> 1 шаг при использовании внутренних операндов, 2 шага в случае использования непосредственно адресуемых выходов DY и 3 шага при использовании всех прочих операндов (включая регистры файлов с последовательным доступом).

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
d	включаемый и выключаемый выходной контакт	бит

**Принцип действия**

**Выходные команды**

**OUT Команда установки выходов**

Выход включается в зависимости от предшествующего входного условия.

За одним и тем же входным условием можно параллельно запрограммировать несколько команд OUT.

Коммутационное состояние контакта OUT можно использовать в качестве входного условия в следующих шагах программы – как состояние обычного замыкающего или размыкающего контакта.

Входное условие	Команда OUT			Если управляется бит словного операнда
	Выходной контакт	Тип контакта		Управляемый бит
		замыкающий	размыкающий	
0	выключен	не проводит	проводит	0
1	включен	проводит	не проводит	1

**Источники ошибок**

См. часть I этого руководства.

**Пример 1**

**OUT**

Следующий пример поясняет программирование команды OUT, если в качестве выходов используются битовые операнды (с Y33 по Y35).

<p>Язык IL для MELSEC</p> <pre> MELSEC LD      X5 OUT     Y33 LD      X6 OUT     Y34 OUT     Y35                     </pre>	<p>Релейно-контактная схема (язык LD)</p>	<p>Язык IL по IEC</p> <pre> LD      X5 ST      Y33 LD      X6 ST      Y34 ST      Y35                     </pre>
---	---	--

**Пример 2**

**OUT**

Следующий пример поясняет программирование команды OUT, если в качестве выходов используются биты словного операнда D0 (биты с b5 по b7).

<p>Язык IL для MELSEC</p> <pre> MELSEC LD      X5 OUT     D0.5 LD      X6 OUT     D0.6 OUT     D0.7                     </pre>	<p>Релейно-контактная схема (язык LD)</p>	<p>Язык IL по IEC</p> <pre> LD      X5 ST      D0.5 LD      X6 ST      D0.6 ST      D0.7                     </pre>
<p>b15 ----- b7b6b5 ---- b0</p> <p>D0 <span style="border: 1px solid black; display: inline-block; width: 100px; height: 15px; vertical-align: middle;"></span></p>		



### 5.3.2 OUT T, OUTH T

**Процессор**

AnS	AnN	An(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

	Операнды														Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки						
	Битовые операнды							Словные операнды (16 бит)										Константы		Указатели		Уровень		M9012	M9011
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1				Z	V	K	H (16#)	P	I	N	
d							●									●	●								
● <sup>1</sup>									● <sup>2</sup>						● <sup>2</sup>	● <sup>2</sup>									

<sup>1</sup> Настройка

<sup>2</sup> Применение расширенных таймеров и счетчиков процессора AnA, AnAS или AnU см. в разд. A.3.4 „Сравнение таймеров“ и разд. A.3.5 „Сравнение счетчиков“ этого руководства.

**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непос. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K			Иные
	битовые	словные		битовые	словные						
d	● <sup>1</sup>	—	—	—	—	—	—	—	—	—	
● <sup>2</sup>	—	● <sup>3</sup>	●	—	●	●	—	● <sup>4</sup>	—	—	

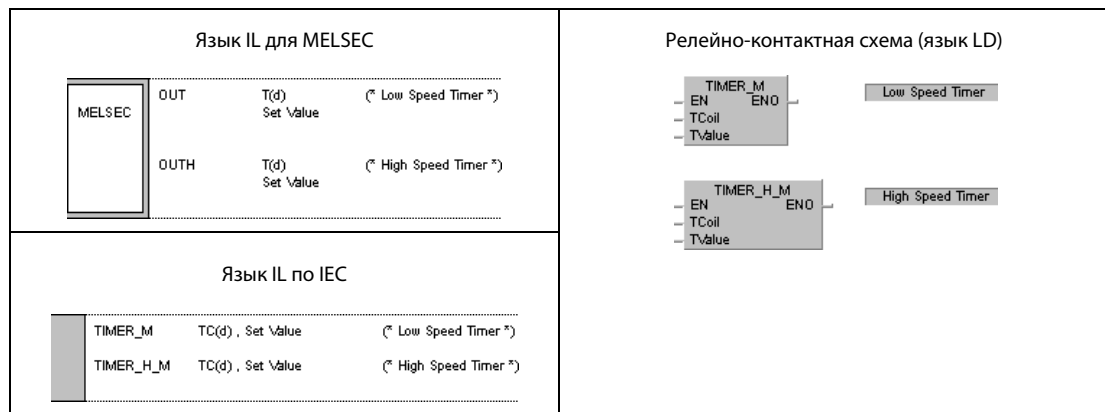
<sup>1</sup> только T

<sup>2</sup> настройка

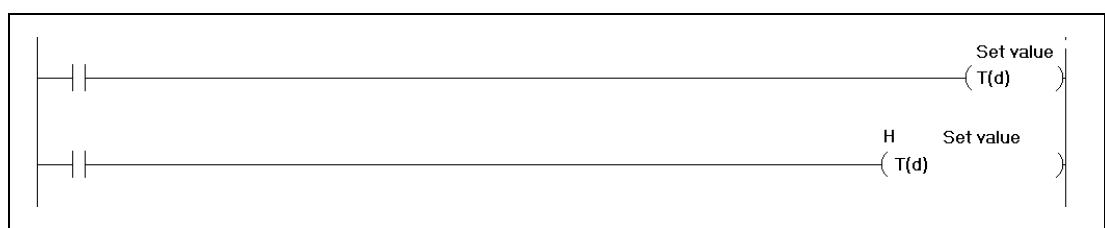
<sup>3</sup> кроме T и C

<sup>4</sup> Настройка для таймера указывается в виде десятичной константы (K). Шестнадцатеричные константы использовать не возможно.

**GX IEC Developer**



**GX Developer**



## Переменные

Операнд	Значение	Тип данных
d	Номер таймера	бит
Настройка	Настройка (Set Value) для таймера	BIN, 16 бит

## Принцип действия

**Команда установки контактов таймера****OUT T медленный таймер (100 мс)****OUTH T быстрый таймер (10 мс)**

Если входное условие команды OUT(H) T выполняется, контакт таймера включается и остается включенным в течение заданного интервала. Этот интервал задается непосредственно с помощью константы либо посредством переменной (содержимого регистра данных).

Коммутационное состояние контакта OUT(H) T используется как входное условие одного или нескольких последующих шагов программы, при этом оно аналогично обычному замыкающему или размыкающему контакту.

По окончании заданного времени (фактическое значение = заданное значение) устанавливается следующий входной контакт.

За одним и тем же входным условием можно запрограммировать несколько команд OUT(H) T.

Таймер в качестве выходного контакта			Таймер в качестве входного условия			
Тип	Состояние контакта	Фактическое значение	Состояние контакта перед истечением установленного времени		Состояние контакта после истечения установленного времени	
			замыкающий	размыкающ.	замыкающий	размыкающ.
100 мс	выключен	0	не проводит	проводит	не проводит	проводит
10 мс			не проводит	проводит	не проводит	проводит
100 мс (фиксируемый)	выключен	фактическое значение сохраняется	не проводит	проводит	проводит	не проводит
10 мс (фиксируемый)			не проводит	проводит	проводит	не проводит

По истечении времени таймера состояние контакта фиксируемого таймера сохраняется до тех пор, пока оно не будет сброшено командой RST.

В качестве заданных значений для таймера нельзя программировать отрицательные значения (от -32768 до -1). Если в качестве заданного значения введен 0, таймер работает так же, как если бы была введена 1.

Выполнение команды OUT(H) T проявляется следующим образом:  
катушка таймера, указанная в d таймера, устанавливается или сбрасывается.  
Соответствующий контакт таймера устанавливается или сбрасывается.  
Значения настройки таймера обновляются.

Если переход к команде OUT(H) T произошел во время ее выполнения, состояния контактов и настройки таймера не изменяются.

Если одна и та же команда применяется в цикле многократно, то при этом обновляется число повторений.

Для индексной адресации катушек счетчиков и контактов могут использоваться только индексные регистры Z0 и Z1.

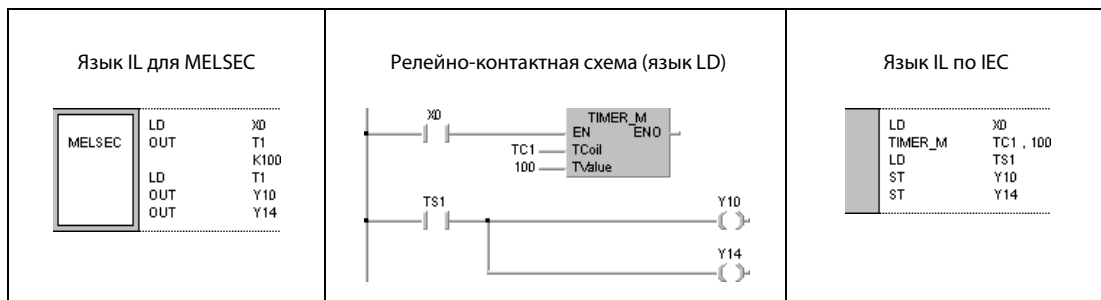
## ПРИМЕЧАНИЯ

*Регистр для заданного значения таймера нельзя адресовать косвенно!*

*Прочие подробности о программировании и принципе действия таймеров можно найти в разд. А.3.4 „Сравнение таймеров“ этого руководства.*

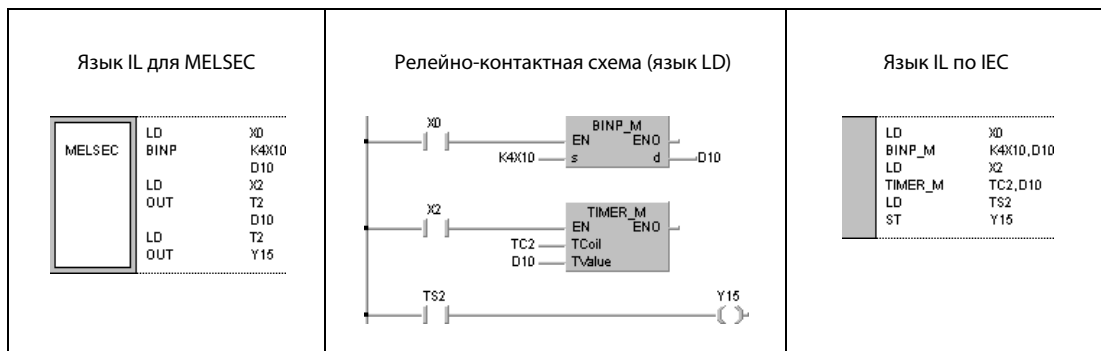
**Пример 1** OUT T

В следующем примере программы через 10 секунд после включения входа X0 устанавливаются выходы Y10 и Y14. Здесь используется медленный таймер (100 мс).



**Пример 2** OUT T

В следующей программе настройка таймера считывается через входы с X10 по X1F в виде двоично-десятичных данных. Для этого сначала двоично-десятичные данные при положительном фронте X0 преобразуются в двоичные данные и сохраняются в D10. После включения X2 настройка таймера считывается. По истечении времени таймера устанавливается Y15. Здесь также используется медленный таймер (100 мс).



**Пример 3** OUTH T

В следующей программе через 250 мс после включения входа X10 устанавливается выход Y10. Здесь применяется быстрый таймер (10 мс).



5.3.3 OUT C

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

	Операнды																Длина блока	Шагов	Индекс	Флаг							
	Битовые операнды								Словные операнды (16 бит)											Константы		Указа-тели		Уро-вень		пере-носа	ошиб-ки
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V				K	H (16#)	P	I	N	M9012	M9010 M9011	
d								●								●	●										
● <sup>1</sup>									● <sup>2</sup>							● <sup>2</sup>	●										

<sup>1</sup> Настройка

<sup>2</sup> Применение расширенных таймеров и счетчиков процессора AnA, AnAS или AnU см. в разд. A.3.4 „Сравнение таймеров“ и разд. A.3.5 „Сравнение счетчиков“ этого руководства.

Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K			Иные U
	битовые	словные		битовые	словные						
d	● <sup>1</sup>	—	—	—	—	—	—	—	—	—	4
● <sup>2</sup>	—	● <sup>3</sup>	●	—	●	●	—	● <sup>4</sup>	—	—	4

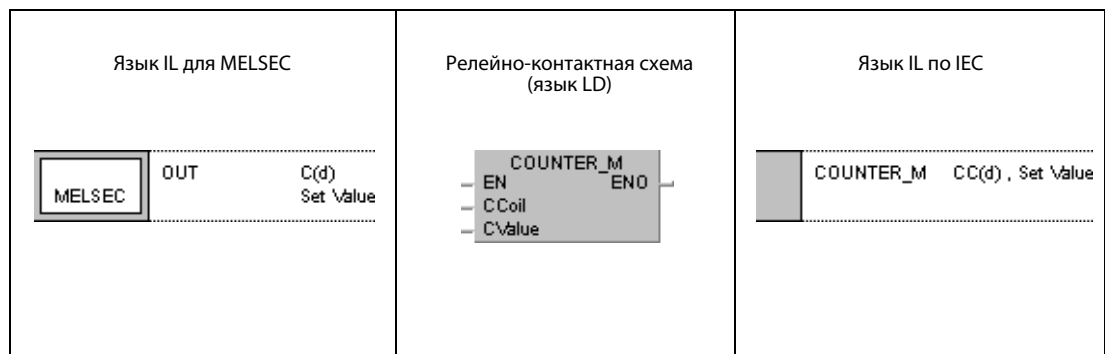
<sup>1</sup> только C

<sup>2</sup> настройка

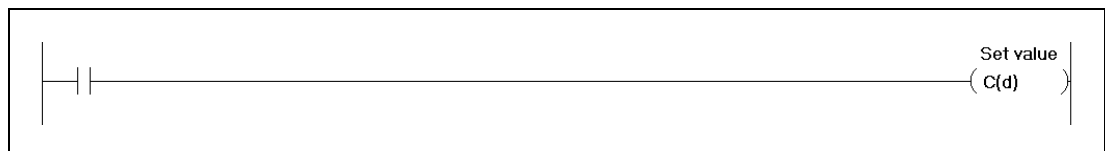
<sup>3</sup> кроме T и C

<sup>4</sup> Настройку для счетчика можно задать только в виде десятичной константы (K). Использование шестнадцатеричных констант (H) или чисел типа REAL не возможно.

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
d	Номер счетчика	бит
Настройка	Настройка (Set Value) для счетчика	BIN, 16 бит

## Принцип действия

### Установка контакта счетчика

#### OUT C счетчик

Если входное условие команды OUT C выполняется, фактическое значение счетчика (значение счета) повышается на 1.

Коммутационное состояние контакта OUT C используется в качестве входного условия в одном или нескольких следующих шагах программы. При этом оно аналогично обычному замыкающему или размыкающему контакту.

Если счетчик достиг заданного значения, устанавливается следующий входной контакт.

Если входное условие команды OUT C остается включенным, процесс счета не продолжается. Поэтому нет никакой необходимости в том, чтобы вход счетчика был импульсным.

По окончании счета значение счета и состояние контакта можно сбросить лишь путем выполнения команды RST.

Если применяются расширенные счетчики с C256 по C1023 для процессоров AnA, AnAS или AnU, заданные значения счетчиков следует запрограммировать в соответствии с разделом "Заданные значения расширенных таймеров и счетчиков" этого руководства.

В качестве заданных значений счетчиков не могут использоваться отрицательные значения (от -32768 до -1). Если в качестве заданного значения введен 0, счетчик работает так же, как если бы была введена 1.

Для индексной адресации катушек и контактов счетчиков могут использоваться только индексные регистры Z0 и Z1.

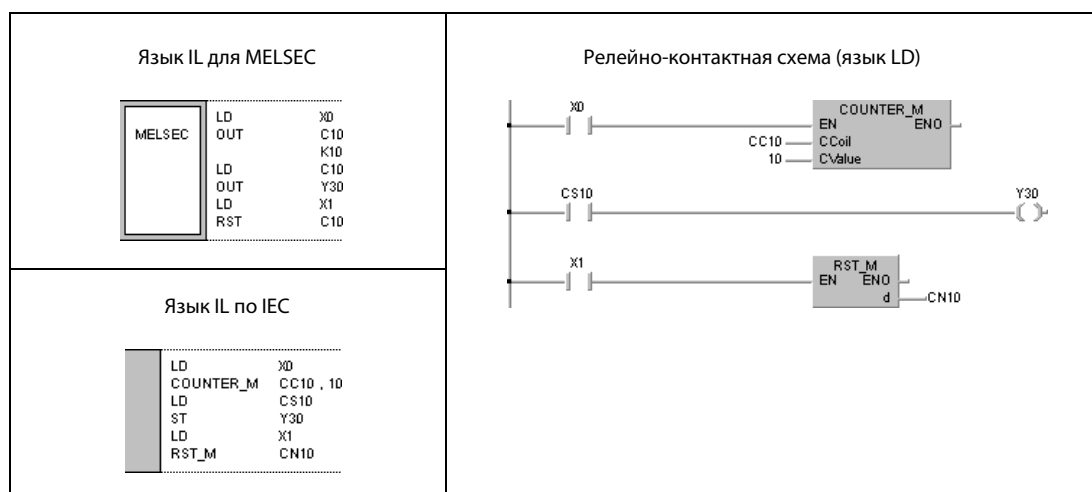
**ПРИМЕЧАНИЯ** Для настроек счетчиков нельзя использовать косвенно адресованные регистры данных.

Прочие подробности о программировании и принципе действия счетчиков можно найти в разд. А.3.5 „Сравнение счетчиков“ этого руководства.

### Пример 1

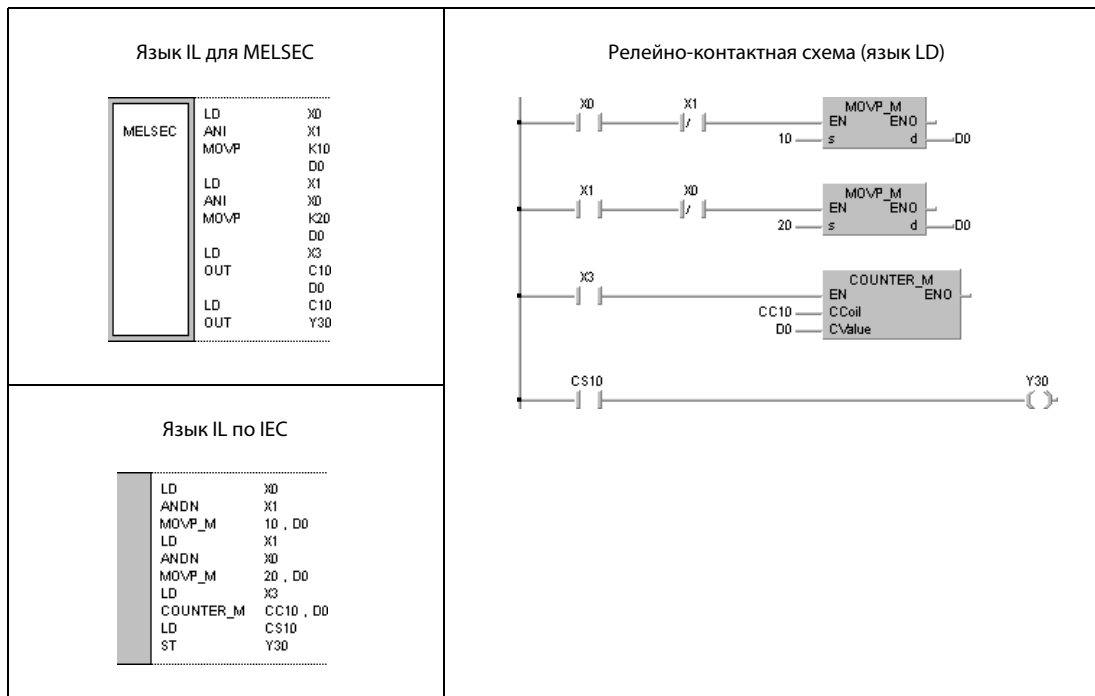
#### OUT C

В следующей программе выход Y30 включается после того, как вход X0 был установлен десять раз, и снова выключается, если устанавливается вход X1.



**Пример 2** OUT C

В следующей программе при положительном фронте сигнала X0 заданное значение счетчика C10 устанавливается на 10 (D0 = 10), а при положительном фронте сигнала X1 – на 20 (D0 = 20). При включении X3 счетчик начинает счет в соответствии с заданным значением в D0. Как только счетчик достигает заданного значения, включается Y30.



### 5.3.4 OUT F

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

d	Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011						
	Битовые операнды						Словные операнды (16 бит)						Константы		Указатели							Уровень					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						K	H (16#)	P	I	N	
						●																		● <sup>1</sup>	● <sup>2</sup>		

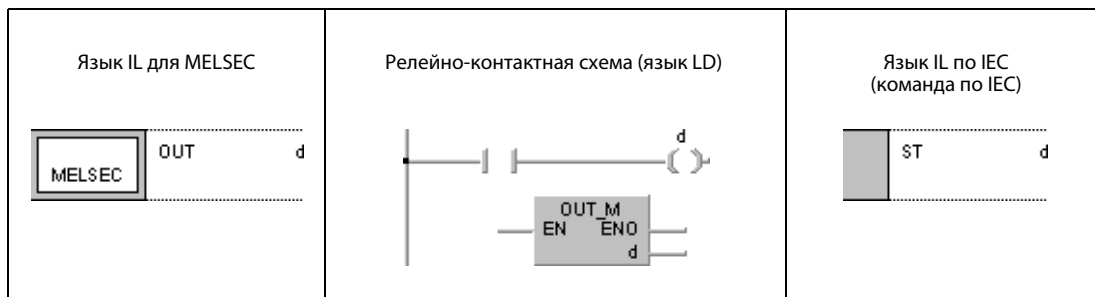
<sup>1</sup> В принципе 1 шаг, однако если в качестве операндов в команде OUT используются специальные маркеры или маркеры ошибки, то расходуются 3 шага. Количество шагов при использовании процессоров AnA, AnAS или AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.  
<sup>2</sup> Функция индекса имеется только для процессоров AnA, AnAS и AnU.

**Операнды MELSEC Q**

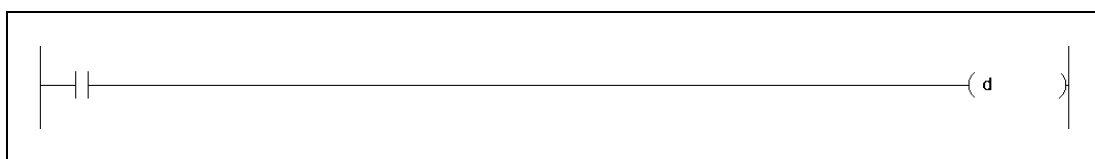
d	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
	● <sup>1</sup>	—	—	—	—	—	—	—	—	4	

<sup>1</sup> только F

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
d	Включаемый маркер ошибки	бит (только маркер F)

**Принцип действия****Вывод маркеров ошибки****OUT F Маркеры ошибки (серии "Q" и "System Q")**

Если входное условие команды OUT F выполняется, включается маркер ошибки и происходит следующий процесс:

Адрес маркера ошибки выводится на светодиодный дисплей центрального процессора (Q3A и Q4AR) и загорается светодиод "USER".

Адреса включенных маркеров ошибки сохраняются в регистрах диагностики с SD64 по SD79.

Значение в SD63 повышается на 1.

Если в регистре диагностики SD63 достигнуто значение 16, т. е., в память записаны 16 адресов включенных маркеров ошибки, то в области от SD64 до SD79 дальнейшие адреса не сохраняются.

Если маркер ошибки выключен с помощью команды OUT, это никак не влияет на светодиодный дисплей, на состояние светодиода "USER" и на содержимое регистров диагностики с SD63 по SD79.

Для стирания маркеров ошибки, регистров и индикации используется команда RST F.

**OUT F Маркеры ошибки (серия "A")**

Если в программе установлен маркер ошибки (F), то на модуле центрального процессора загорается светодиод "ERROR" и соответствующие светодиодные индикаторы. Количество включенных маркеров ошибки сохраняется в специальном регистре.

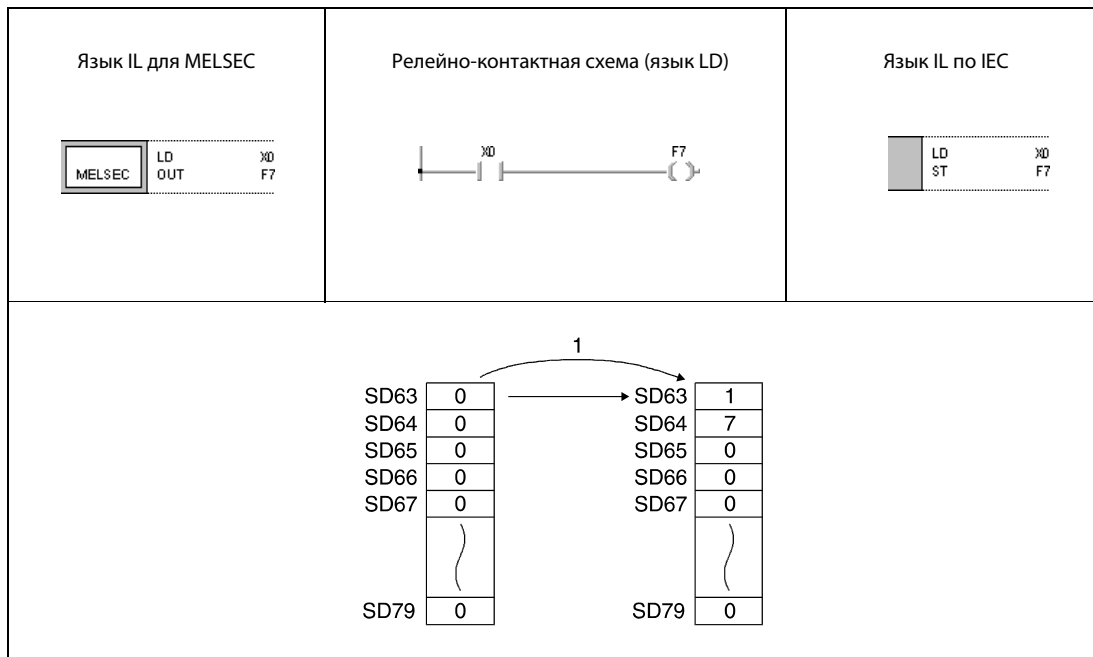
Маркеры ошибки нельзя устанавливать с помощью команды OUT, так как светодиодная индикация ошибок не совпадает с состоянием контакта выходной команды. Во избежание этого несовпадения маркеры ошибки следует устанавливать с помощью команды SET. Включение маркера ошибки с помощью команды OUT приводит также к тому, что при исчезновении входного условия выключается и маркер ошибки. На состояние светодиодной индикации, состояние светодиода "ERROR" на модуле центрального процессора и содержимое специальных регистров это не влияет.



**Пример  
(серия "Q")**

**OUT F**

В следующей программе при включении входа X0 включается маркер ошибки F7. Значение 7 сохраняется в регистрах с SD64 по SD79. Значение в регистре SD63 повышается на 1 (т. е. сохранен еще один адрес маркера ошибки).



<sup>1</sup> Включается X0.

### 5.3.5 SET

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

Операнды																	Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки				
Битовые операнды							Словные операнды (16 бит)							Константы		Указатели						Уровень			
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K				N (16#)	P	I	N		
d	●	●	●	●	●																	1	1		2

<sup>1</sup> Если с помощью команды SET устанавливается специальный маркер, маркер связи или маркер ошибки (M, B или F) или сбрасывается специальный маркер или любой словный операнд, то количество шагов равно 3.

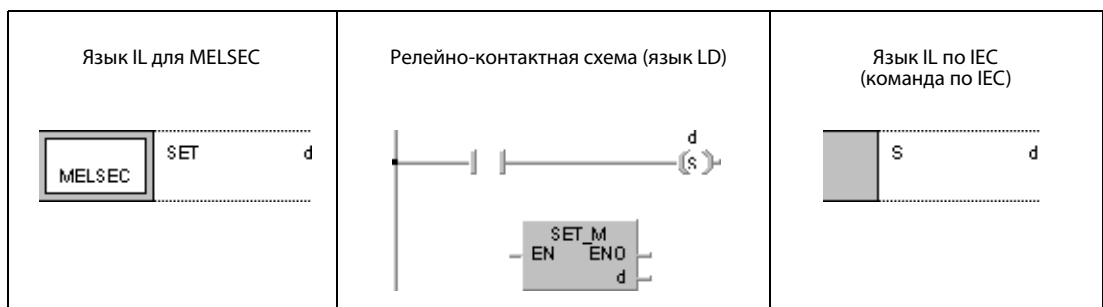
<sup>2</sup> Функция индекса имеется только для процессоров AnA, AnAS и AnU.

**Операнды MELSEC Q**

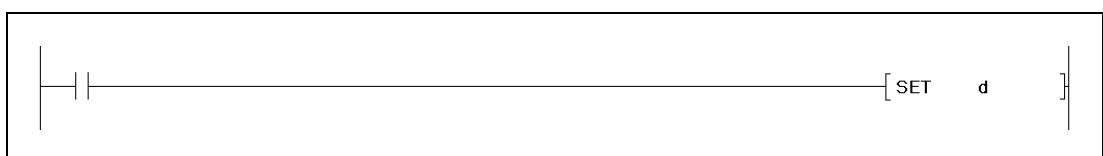
Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непос. адр. J□□		Спец. модули U□G□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные				BL	DY		
d	●	●	●	●	●	—	—	●	●	—	1

<sup>1</sup> 1 шаг при использовании внутренних операндов, 2 шага при использовании непосредственно адресуемых выходов DY или программных операндов SFC (BL), 3 шага при использовании всех прочих операндов (включая регистры файлов с последовательным доступом) и 4 шага при использовании таймеров (T) или счетчиков (C).

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
d	Включаемый выходной контакт или бит словного операнда	бит

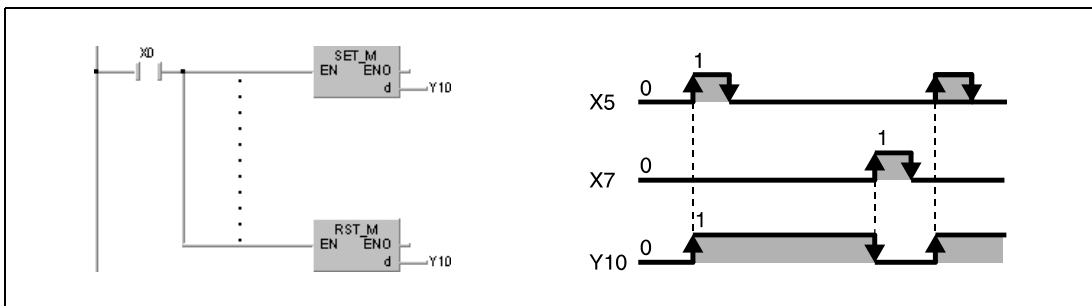
**Принцип действия**

**Установка операнда SET команда установки**

Команда SET состоит из слова SET ("установить"), за которым следует адрес устанавливаемого операнда d.

После выполнения входного условия устанавливается команда SET, а также устанавливается указанный адрес операнда d или определенному биту словного операнда присваивается значение 1.

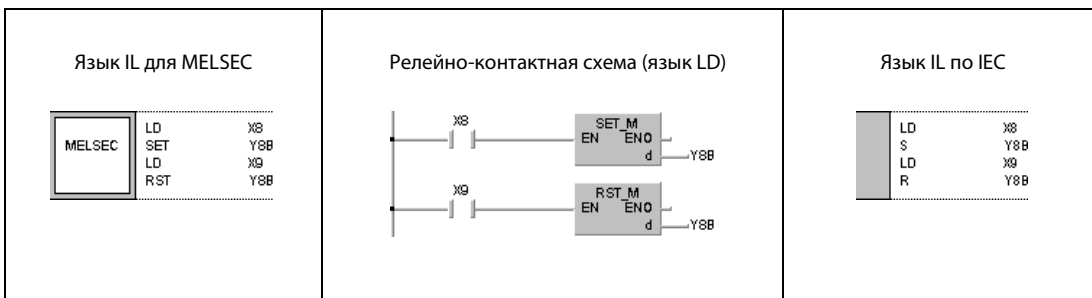
Если входное условие снова выключилось, установленный операнд остается установленным. Его можно сбросить с помощью команды RST.



**Пример 1**

SET

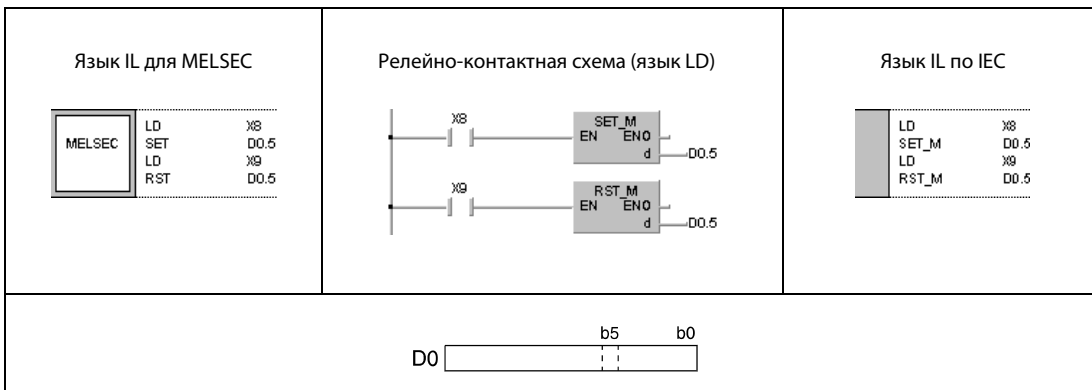
В следующей программе при включении входа X8 устанавливается выход Y8B. При включении X9 выход Y8B сбрасывается.



**Пример 2**

SET

В следующей программе при включении входа X8 состояние бита 5 (b5) в D0 изменяется с 0 на 1. При включении X9 этот бит сбрасывается.



### 5.3.6 RST

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

	Операнды																Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки						
	Битовые операнды						Словные операнды (16 бит)						Константы		Указатели							Уровень					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V							K	H (16#)	P	I	N
d	●	●	●	●	●		●	●	●	●	●	●	●	●	●								1	● <sup>1</sup>	● <sup>2</sup>		

<sup>1</sup> Если с помощью команды SET устанавливается специальный маркер, маркер связи или маркер ошибки (M, B или F) или сбрасываются специальный маркер или любой словный операнд, то количество шагов равно 3.

<sup>2</sup> Функция индекса имеется только для процессоров AnA, AnAS и AnU.

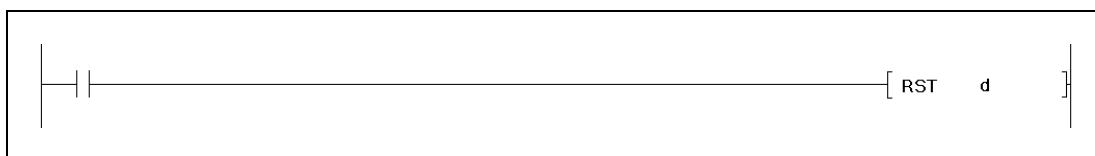
**Операнды MELSEC Q**

	Операнды										Флаг ошибки	Шагов
	Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)	Иные			
	битовые	словные		битовые	словные				BL	DY		
d	●	●	●	●	●	●	●	—	—	●	—	2

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
d	Сбрасываемый выходной контакт или бит словного операнда	бит ● <sup>1</sup>

<sup>1</sup> Особенность команды RST\_M заключается в возможности стирания целых слов данных. При этом расходуетс меньшее количество шагов, чем при применении команды MOV с константой K0.

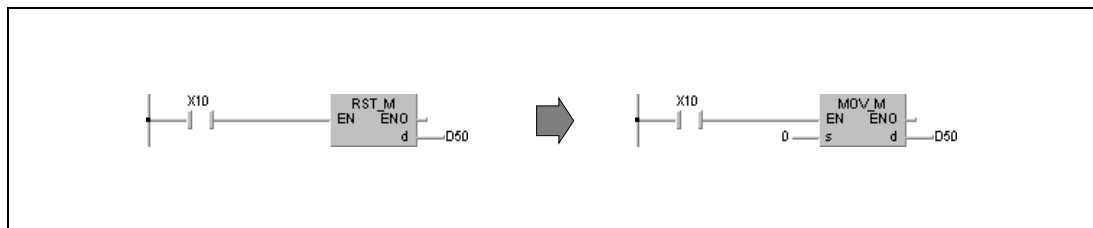
**Принцип действия**

**Сброс операнда  
RST команда сброса**

Команда RST состоит из слова RST (reset, "сброс"), за которым следует адрес сбрасываемого операнда.

После выполнения команды RST входные и выходные контакты битовых операндов отключаются, фактические значения таймеров и счетчиков (T, C) устанавливаются на 0 и их контакты выключаются, установленный бит словного операнда сбрасывается на 0, а также содержимое словных операндов устанавливается на 0.

На следующей иллюстрации команда RST выполняет ту же функцию, что и показанная справа команда MOV. В качестве входа команды RST используется X10.



**Пример 1**

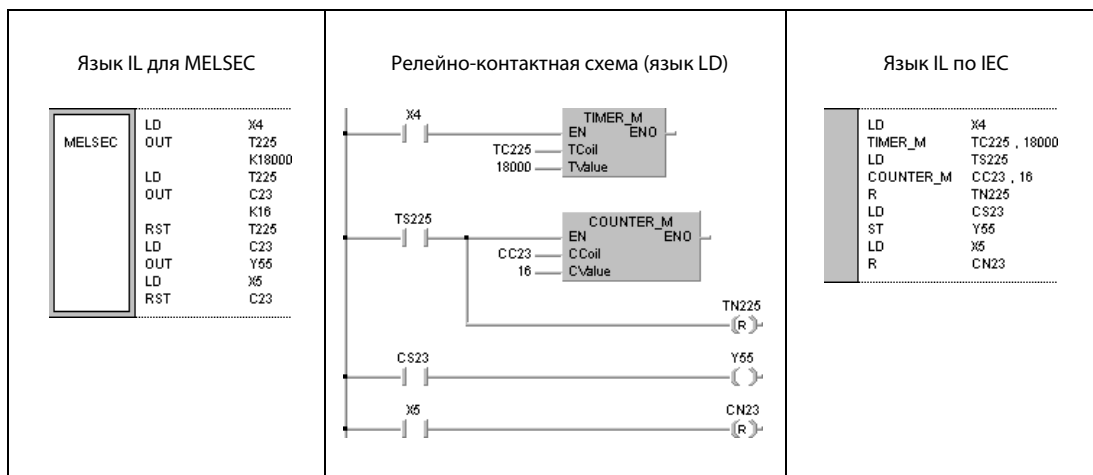
**RST**

В следующей программе при положительном фронте сигнала X0 содержимое адресов с X10 по X1F записывается в регистр данных D8. После включения X5 содержимое D8 сбрасывается на 0.



**Пример 2** RST T, C

В следующей программе показан пример сброса фиксируемых таймеров и счетчиков. В первой строке программы, если X4 включен в течение 30 минут, устанавливается T225. Во втором шаге моменты включения T225 регистрируются счетчиком C23. Одновременно сбрасывается T225. После того, как таймер был включен 16 раз (фактическое значение C23 = 16), включается выход Y55. После включения X5 счетчик устанавливается на 0.



### 5.3.7 SET F, RST F

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

d	Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011			
	Битовые операнды						Словные операнды (16 бит)						Константы		Указатели							Уровень		
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						K	H (16#)	P
						●															3	● <sup>1</sup>		

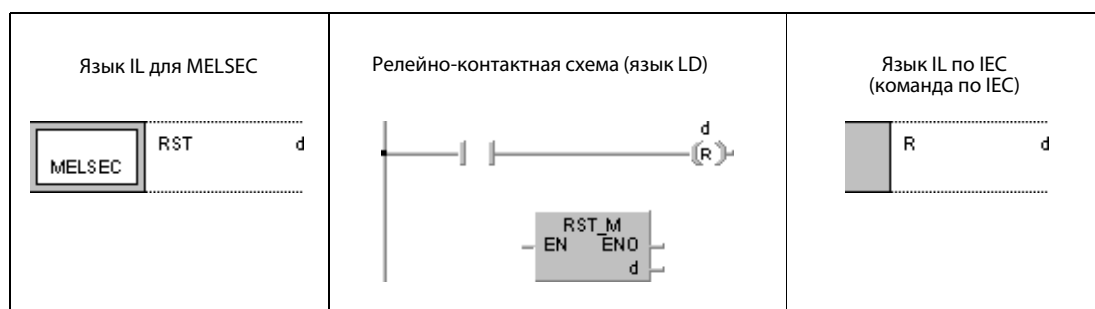
<sup>1</sup> Функция индекса имеется только для процессоров AnA, AnAS или AnU.

**Операнды MELSEC Q**

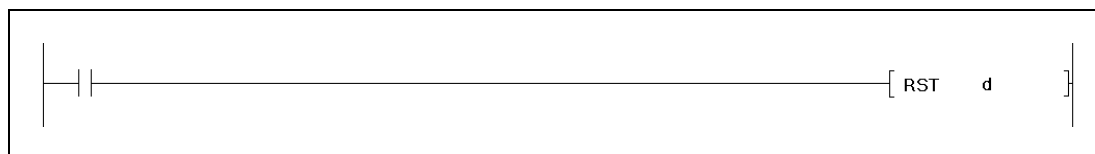
d	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□\G□□	Индексные регистры Zn	Константы K, H (16#)			Иные U
	битовые	словные		битовые	словные						
	● <sup>1</sup>	—	—	—	—	—	—	—	—	3	

<sup>1</sup> только F

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
d (SET)	Устанавливаемый маркер ошибки	бит (только F)
d (RST)	Сбрасываемый маркер ошибки	бит (только F)

**Принцип действия****Установка и сброс маркеров ошибки (серии "Q" и "System Q")****SET F    Команда установки**

Команда SET F состоит из слова SET ("установить"), за которым следует адрес устанавливаемого операнда d. После выполнения условия входа устанавливается команда SET и указанный адрес операнда d. Для установки маркера ошибки команда SET выводит импульс.

Происходит следующий процесс:

Адрес маркера ошибки выводится на светодиодный дисплей центрального процессора (Q3A и Q4AR) и загорается светодиод "USER".

Адреса включенных маркеров ошибки сохраняются в регистрах с SD64 по SD79.

Значение в SD63 повышается на 1.

Если в регистре SD63 достигнуто значение 16, т. е., записаны уже 16 адресов включенных маркеров ошибки, в области от SD64 до SD79 более не сохраняются никакие адреса.

**RST F    Команда сброса**

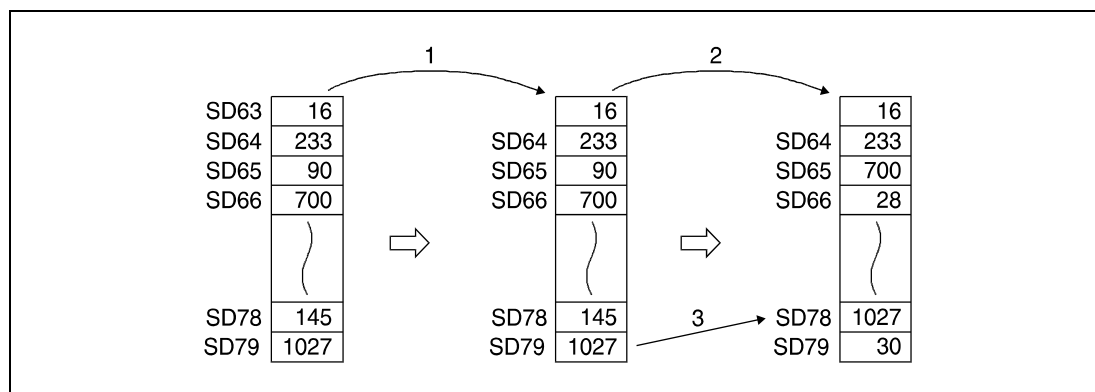
Команда RST F состоит из слова RST ("reset", "сброс"), за которым следует адрес устанавливаемого операнда d.

После выполнения условия входа устанавливается команда RST, и указанный адрес операнда d сбрасывается. Выходной сигнал для сброса маркера ошибки является импульсным.

Адрес выключенного маркера ошибки стирается из регистров с SD64 по SD79 и значение в регистре SD63 понижается на 1. Если значение в регистре SD63 было равным 16 и команда RST стерла маркеры ошибки из регистров с SD64 по SD79, то теперь включаются те маркеры ошибки, которые перед этим не могли быть сохранены. Адреса этих маркеров ошибки сохраняются в освободившихся регистрах в диапазоне между SD64 и SD79.

Если после выполнения команды RST F значение в регистре диагностики SD63 снизилось до 0 и все маркеры ошибки выключились, светодиодный дисплей и светодиод "USER" гаснут.

На нижней иллюстрации в первом шаге (1) программы устанавливается маркер F30, однако он не может быть записан в регистр, так как в области от SD64 до SD79 уже сохранены 16 адресов. Во втором шаге (2) сбрасывается F90. В результате этого становится возможным сохранение маркера F30 в SD79, так как прочие сохраненные маркеры ошибки сдвигаются вверх на один освободившийся регистр (SD65) (3).





**Установка и сброс маркеров ошибки (серия "А")**

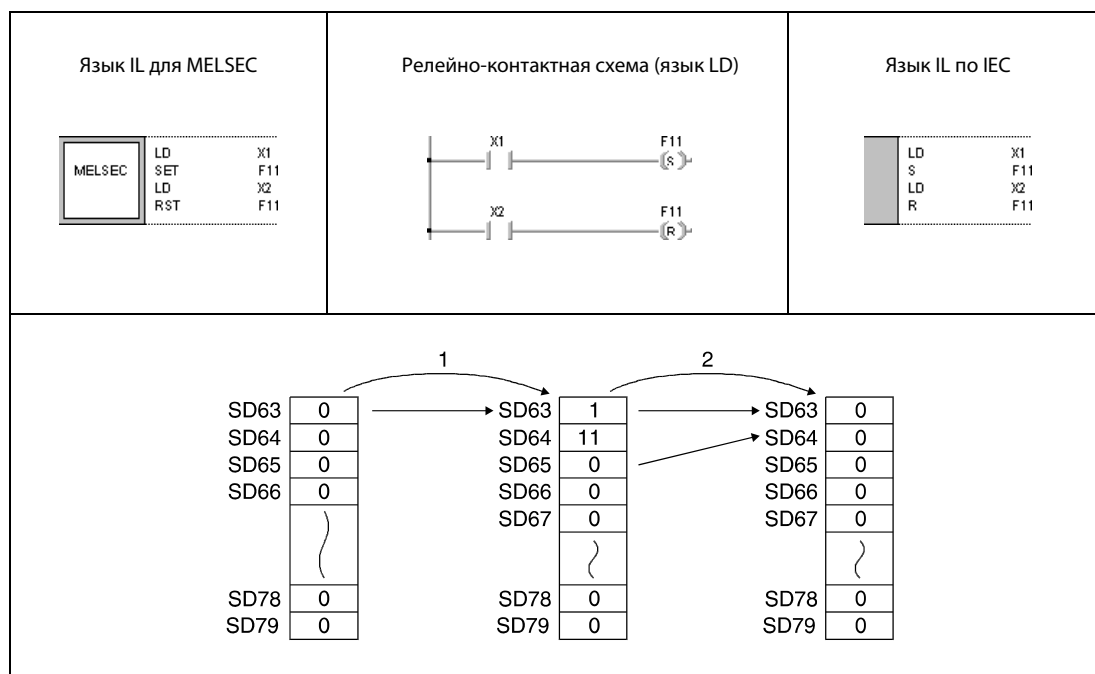
**SET F/ RST F      Команда установки/сброса**

Если с помощью команды SET (RST) установлен (сброшен) маркер ошибки F, то в результате этого изменяется состояние соответствующих светодиодных индикаторов (дисплеев) и светодиода "Error" на модуле центрального процессора, а также содержимое соответствующих специальных регистров. Маркеры ошибки устанавливаются и сбрасываются импульсными сигналами.

**Пример**

SET F/RST F (серии "Q" и "System Q")

В следующей программе включается маркер ошибки F11, если устанавливается вход X1 (1). Значение 11 сохраняется в регистрах SD64...SD79, и значение в SD63 повышается на 1. После этого маркер ошибки F11 выключается, если устанавливается вход X2 (2). Значение 11 стирается из регистров диагностики SD64...SD79, и значение в SD63 уменьшается на 1.



5.3.8 PLS, PLF

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

	Операнды															Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки						
	Битовые операнды						Словные операнды (16 бит)						Константы		Указатели						Уровень					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z				V	K	H (16#)	P	I	N	M9012	M9010 M9011
d	●	●	●	●	●	●																	3	● <sup>1</sup>	● <sup>2</sup>	

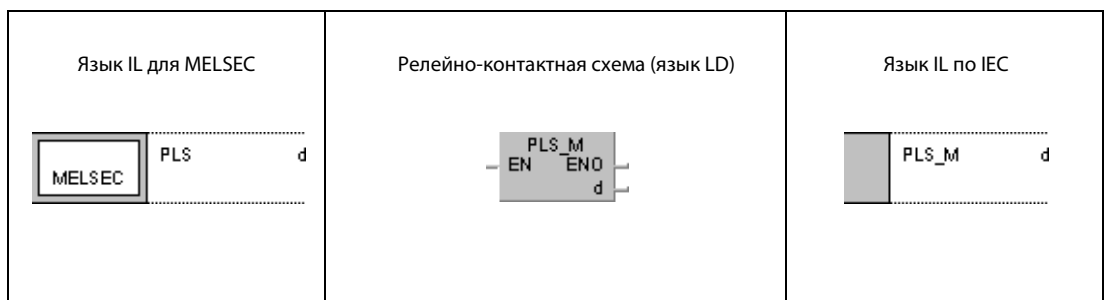
<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

<sup>2</sup> Функция индекса имеется только для процессоров AnA, AnAS или AnU.

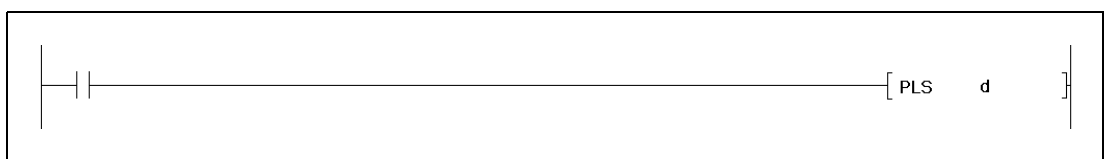
Операнды MELSEC Q

	Операнды										Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные			
	битовые	словные		битовые	словные				DY			
d	●	●	●	●	●	●	—	—	●	—	—	2

GX IEC Developer



GX Developer



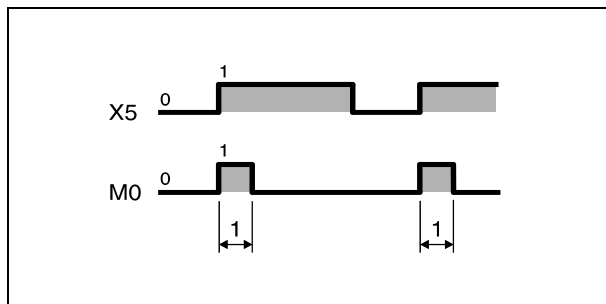
Переменные

Операнд	Значение	Тип данных
d	Операнд, выходной сигнал которого преобразуется в импульс.	бит

**Принцип действия****Вывод по фронту сигнала****PLS Вывод при положительном фронте сигнала**

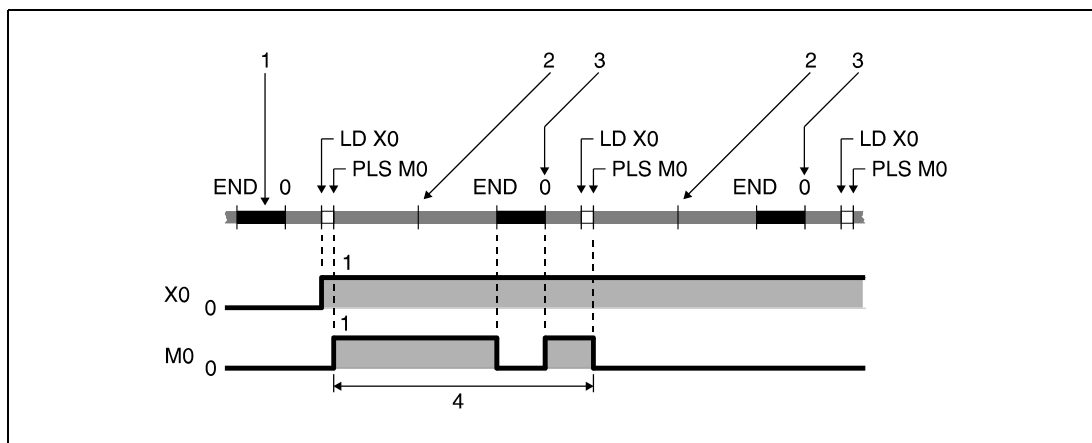
Команда PLS состоит из слова PLS (т. е. "pulse", "импульс") и следующего за ним адреса устанавливаемого операнда d.

При нарастающем (положительном) фронте сигнала входного условия команда PLS включает операнд на время одного программного цикла. Если заданный операнд уже установлен, то на один программный цикл этот операнд выключается.



<sup>1</sup> один цикл

Если команда, вырабатывающая импульс, установлена, и после этого переключатель режимов центрального процессора переключен с "RUN" на "STOP", а затем снова на "RUN", команда PLS не выполняется.



<sup>1</sup> обработка команды END

<sup>2</sup> переключатель режимов "RUN" → STOP

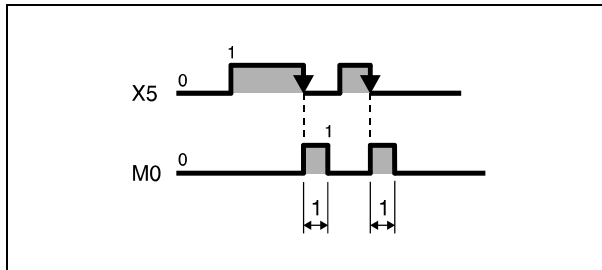
<sup>3</sup> переключатель режимов "STOP" → RUN

<sup>4</sup> один цикл PLS M0

Если в сочетании с командой PLS запрограммирован фиксируемый маркер L и перед отключением центрального процессора этот фиксируемый маркер был установлен, то в первом цикле после повторного включения центрального процессора команда PLS вырабатывает импульс. В следующем цикле управляемый командой PLS операнд сбрасывается.

**PLF Вывод при отрицательном фронте сигнала**

Команда PLF состоит из слова PLF и следующего за ним адреса устанавливаемого операнда d. При понижающемся (отрицательном) фронте сигнала входного условия команда PLF включает операнд на время одного программного цикла. Если заданный операнд уже установлен, то на один программный цикл этот операнд выключается.



<sup>1</sup> один цикл

Если команда, вырабатывающая импульс, установлена, и после этого переключатель режимов центрального процессора переключен с "RUN" на "STOP", а затем снова на "RUN", то команда PLF не выполняется.

**ПРИМЕЧАНИЕ** Если команда PLS или PLF запрограммирована как цель команды перехода (например, CJ) и эта часть программы не выполняется, то указанный в команде PLS или PLF операнд остается установленным более одного программного цикла.

**Пример 1** PLS

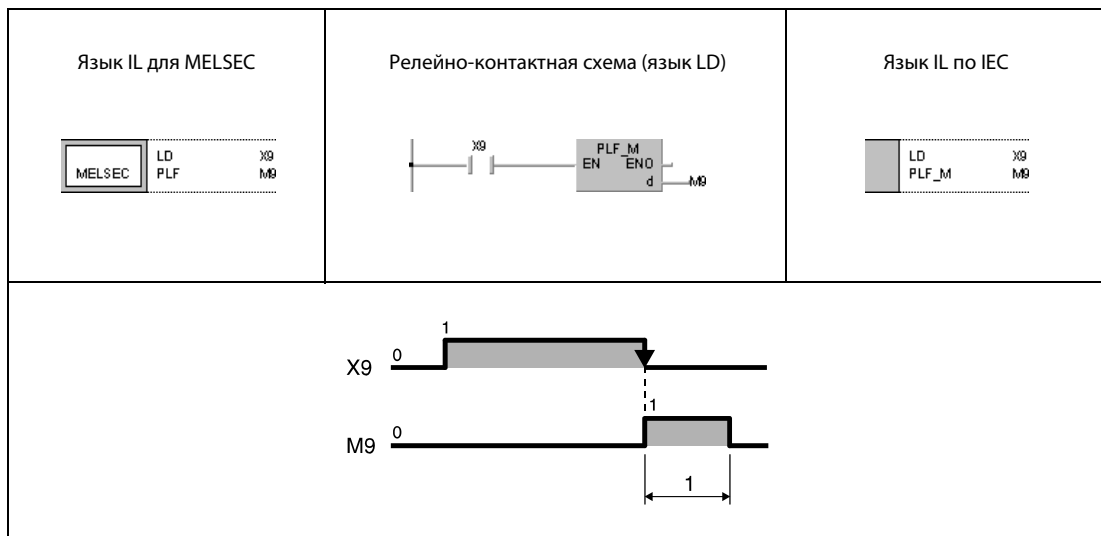
В следующей программе при положительном фронте сигнала X9 на один программный цикл устанавливается маркер M9.

Язык IL для MELSEC	Релейно-контактная схема (язык LD)	Язык IL по IEC

<sup>1</sup> один цикл

**Пример 2** PLF

В следующей программе при отрицательном фронте сигнала X9 на один программный цикл устанавливается маркер M9.



<sup>1</sup> один цикл

5.3.9 FF

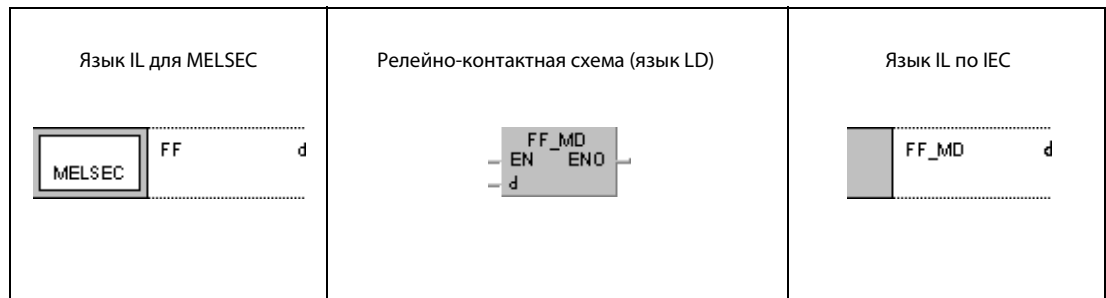
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

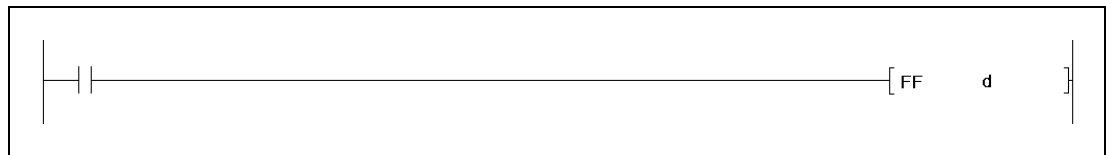
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные DY
	битовые	словные		битовые	словные						
d	●	●	●	●	●	●	—	—	●	—	2

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
d	Битовый операнд или адресованный бит словного операнда, состояние которого инвертируется.	бит

**Принцип действия**

**Изменение состояния битового выходного операнда**

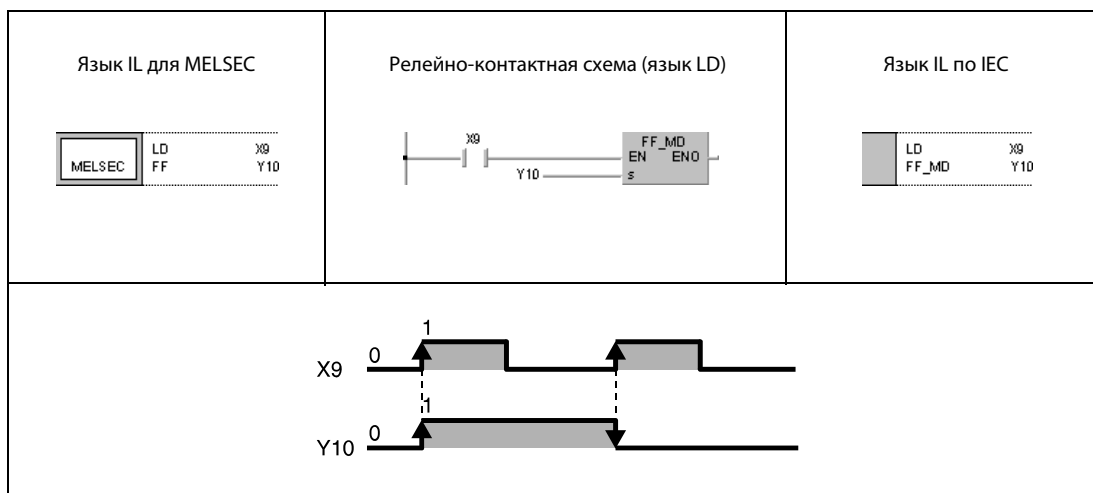
**FF Инвертирование битового выходного операнда**

При положительном фронте на входе команды FF состояние сигнала операнда, указанного в d, изменяется на противоположное. Операндом может быть битовый операнд или определенный бит словного операнда. Если перед этим состояние выходного операнда было равным 1, то после выполнения команды FF оно становится равным 0. Если перед выполнением команды состояние выходного операнда было 0, то после этого оно становится равным 1.

**Пример 1**

FF

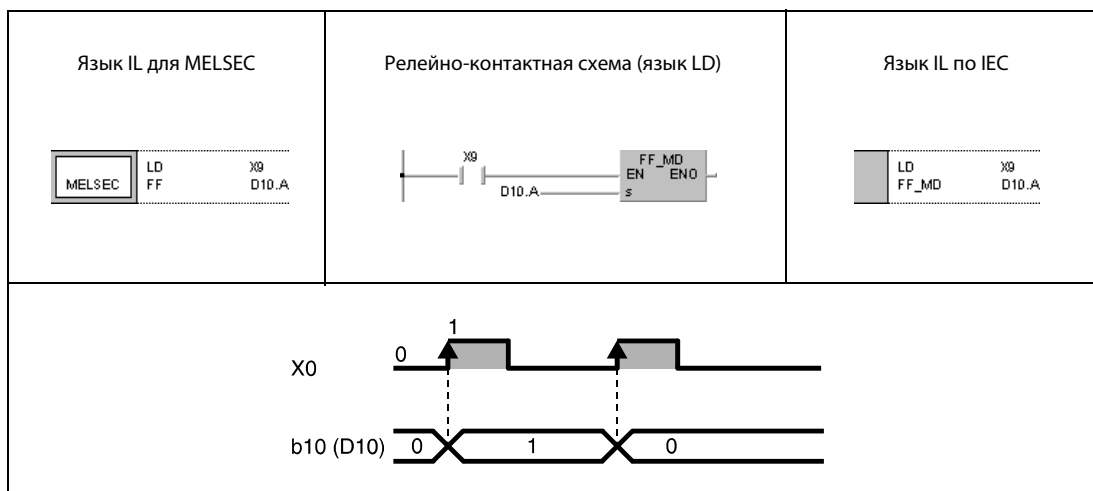
В следующей программе при положительном фронте сигнала X9 инвертируется выходное состояние Y10.



**Пример 2**

FF

В следующей программе при положительном фронте сигнала X9 инвертируется бит 10 (b10) регистра D10.



5.3.10 CHK

Процессор

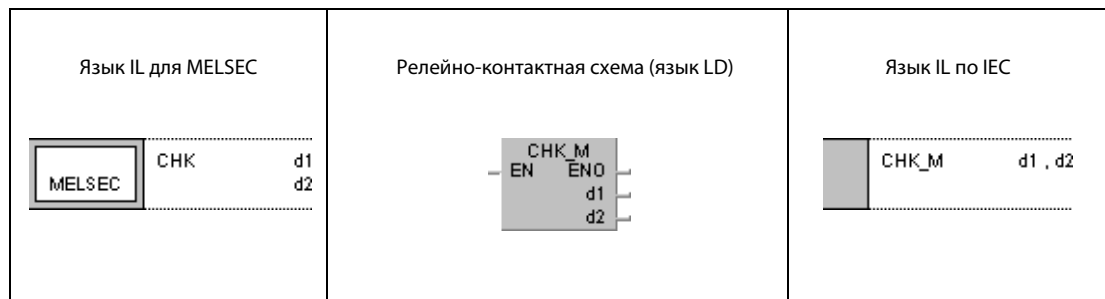
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●				

Операнды MELSEC A

	Операнды																Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки				
	Битовые операнды							Словные операнды (16 бит)								Константы				Указатели		Уровень	M9012	M9010 M9011	
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V				K	H (16#)	P	I	N	
d1	●	●	●	●	●	●																			
d2 1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							K1 ┆ K4	5		

<sup>1</sup> Операнд d2 не влияет на выполнение программы.

GX IEC Developer



Переменные

Операнд	Значение	Тип данных
d1	Битовый операнд, состояние которого инвертируется.	бит
d2	Фиктивный операнд (dummy)	бит



**Принцип действия**

**Изменение состояния битового выходного операнда (серия "А")**

**Общие сведения**

Функция команды CHK зависит от выбранного вида обработки. При непосредственной обработке состояний входов и выходов (кроме процессоров AnA и A2C) эта команда служит для контроля на ошибки. В процессорах AnS и AnN с обновляемым отображением входов и выходов команда CHK позволяет инвертировать коммутационное состояние выходного операнда (Flip-Flop).

**CHK Инвертирование битового выходного операнда**

Полная команда CHK состоит из слова CHK, операнда d1, состояние которого требуется инвертировать, и фиктивного операнда d2.

Если входное условие команды CHK выполнено, состояние указанного в команде CHK операнда изменяется на противоположное. После выключения и повторного включения входного условия указанный операнд снова возвращается в исходное состояние.

Хотя d2 является лишь т. н. "фиктивным" операндом, он должен быть указан (см. таблицу операндов). Если в качестве d2 указан битовый операнд, то с помощью K1...K4 следует установить длину блока. Значение длины блока не играет роли, так как оно мнимое. Указанный в d2 операнд можно свободно использовать и для других целей.

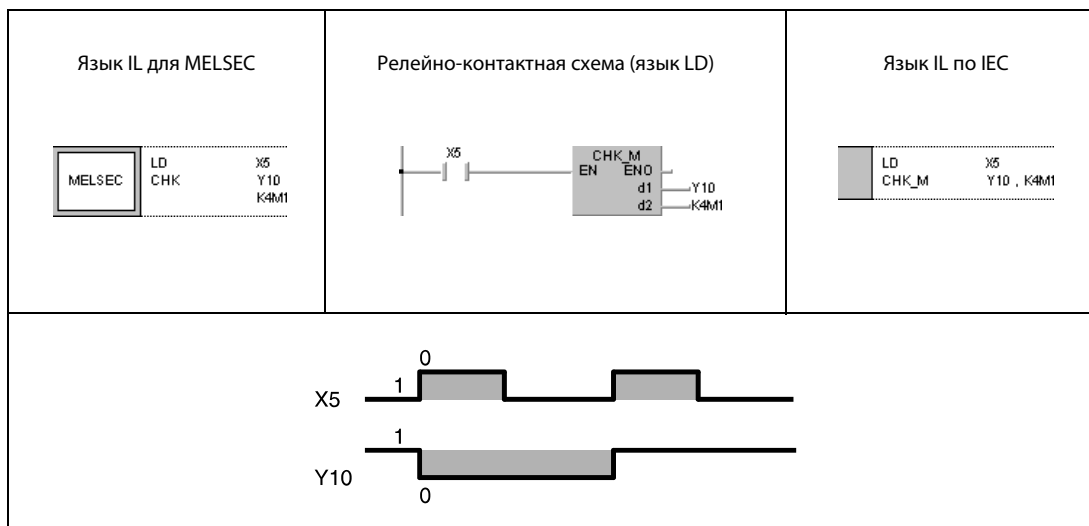
Описываемая здесь команда CHK выполняется только в режиме обновляемого отображения входов и выходов.

Инвертирование состояния выходного операнда должно длиться по меньшей мере один программный цикл.

**Пример**

CHK

В следующей программе при положительном фронте сигнала X5 инвертируется выходное состояние Y10.



5.3.11 DELTA, DELTAP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	Q
				●	●

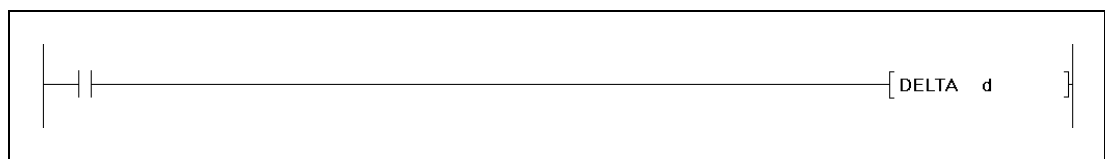
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные DY
	битовые	словные		битовые	словные						
d	—	—	—	—	—	—	—	—	●	SM0	2

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
d	Непосредственно адресуемый выход, в отношении которого применяется команда.	бит ● <sup>1</sup>

<sup>1</sup> только непосредственные выходы

**Принцип действия**

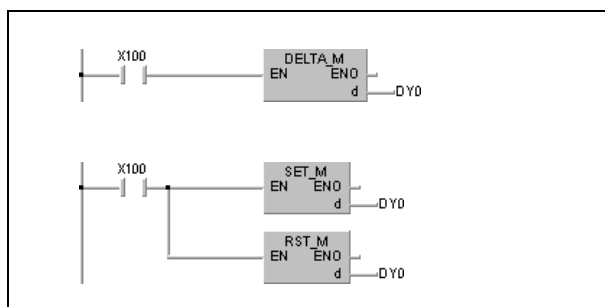
**Выработка импульса переключения на непосредственно адресуемом выходе**

**DELTA Команда для импульсов переключения**

Команды DELTA вырабатывают импульс переключения на указанном в d непосредственно адресуемом выходе (DY), т. е. этот выход устанавливается на определенное время.

Если управляемый командой DELTA выход имеет номер DY0, то функция этой команды идентична функции команд SET/RST (см. рис.).

Команда DELTA применяется в модулях специальных функций для команд, управляемых по фронту сигнала.



**Источники ошибок**

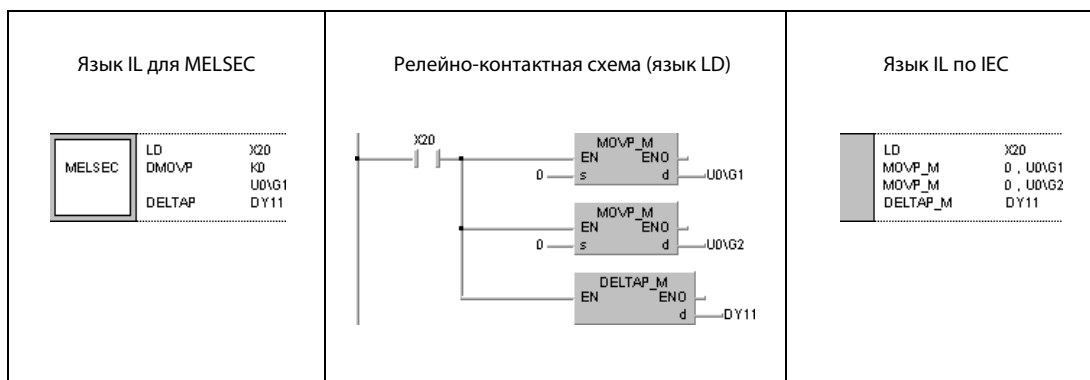
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанный в d выход находится вне диапазона адресов для выходов (код ошибки 4101).

**Пример**

**DELTAP**

В следующей программе при положительном фронте сигнала X20 выполняется предварительная установка CH1 модуля вывода AD61 в слоте 0 главной монтажной шины. Адреса 1 и 2 в буферной памяти модуля вывода предварительно заполняются нулями. При выполнении команды DELTAP активируется эта предварительная установка.



## 5.4 Команды сдвига

### 5.4.1 SFT, SFTP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

Операнды														Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011						
Битовые операнды						Словные операнды (16 бит)						Константы							Указатели		Уровень			
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1						Z	V	K	N (16#)	P	I
d	●	●	●	●	●	●															3 <sup>1</sup>	● <sup>2</sup>		

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

<sup>2</sup> Функция индексации имеется только для процессоров AnA, AnAS и AnU.

Операнды MELSEC Q

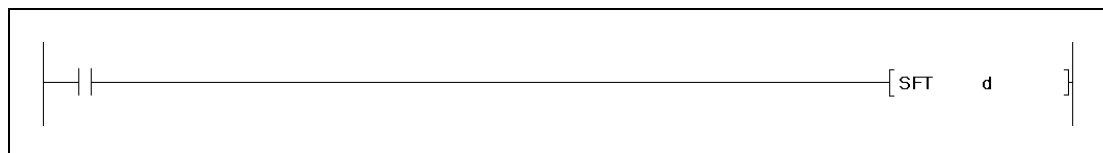
Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы DY	Иные			
битовые	словные		битовые	словные				U			
d	● <sup>1</sup>	● <sup>1</sup>	● <sup>1</sup>	● <sup>1</sup>	● <sup>1</sup>	—	—	●	—	—	2

<sup>1</sup> кроме T и C

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
d	Сдвигаемый операнд	бит

**Принцип действия**

**Команда сдвига**

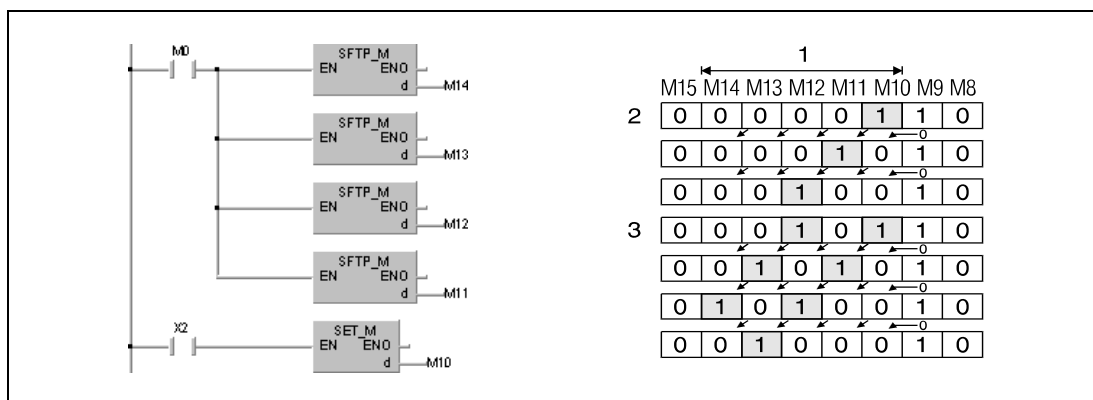
**SFT Сдвиг битовых операндов**

Команда SFT сдвигает операнд на один бит. Сдвиг происходит только при выполнении входного условия команды SFT (положительный фронт).

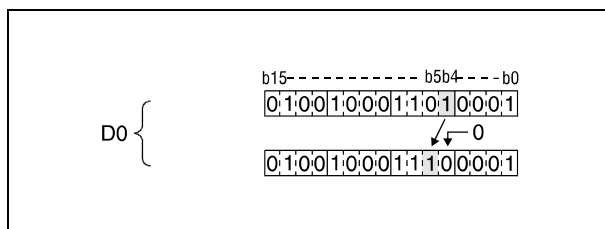
При выполнении команды состояние адреса операнда (указанного в d – 1) перемещается в целевой адрес d. Состояние операнда с более низким адресом d – 1 сбрасывается. Установить операнд сдвинутого адреса можно с помощью команды SET.

При программировании нескольких команд SFT подряд следует начинать с операнда с более высоким адресом.

На рисунке ниже при установке входа X2 (2, 3) устанавливается маркер M10. Состояние M10 (1) сдвигается командой SFT P в пределах области сдвига (1).



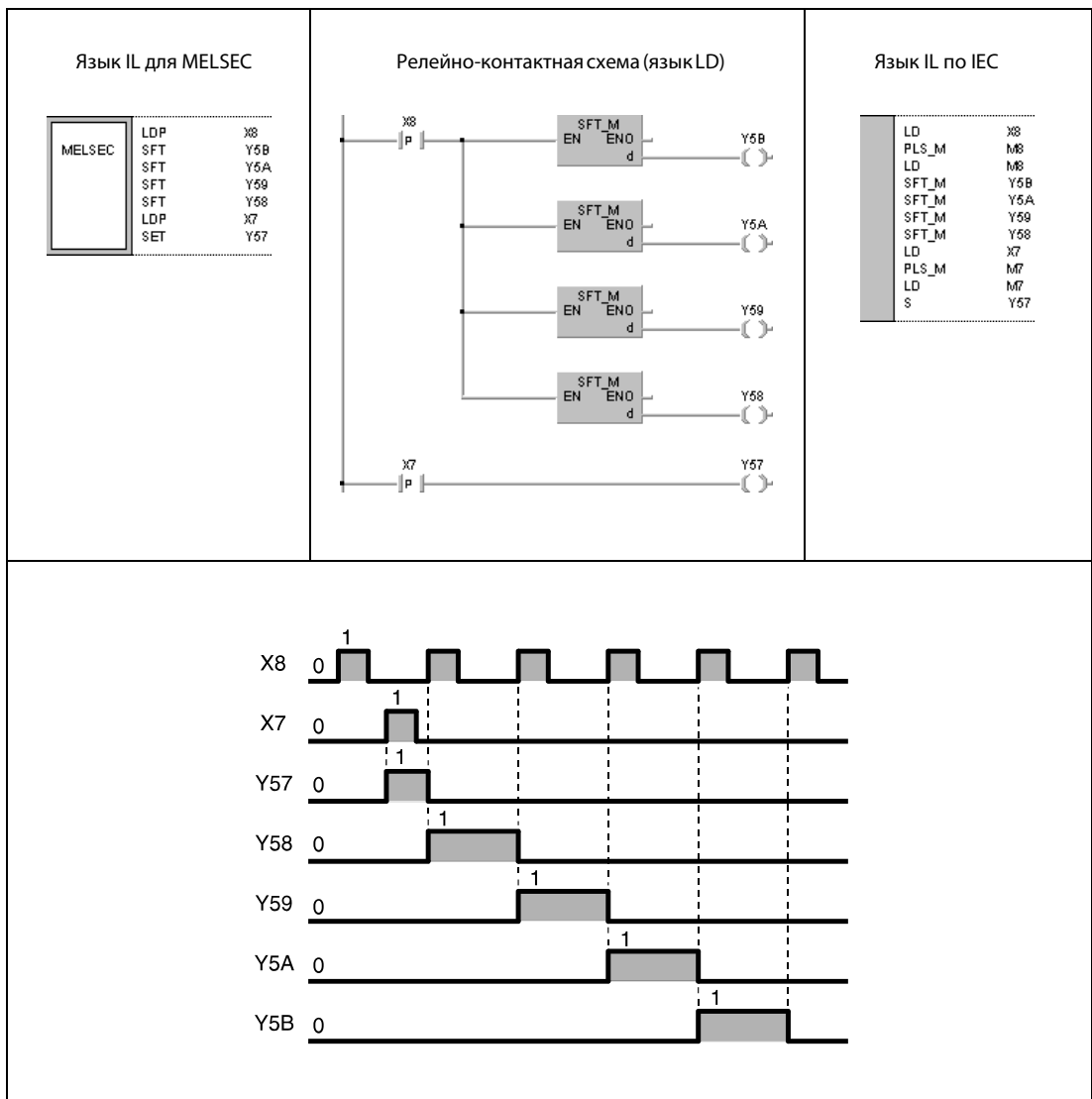
При сдвиге битов в словных операндах состояние (0/1) бита d – 1 перемещается в бит d. После выполнения команды SFT в бит d – 1 записывается 0. На следующей иллюстрации сдвигается бит 5 (b5) в D0. После выполнения команды в бит 4 (b4) записывается 0.



**Пример**

SFT

В следующей программе при положительном фронте сигнала X8 состояние Y57 перемещается в Y5B. Затем Y57 устанавливается при положительном фронте X7.



## 5.5 Организирующие команды

### 5.5.1 MC, MCR

**ПРИМЕЧАНИЕ** Эти команды не следует использовать в редакторах по стандарту IEC.

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

	Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011						
	Битовые операнды						Словные операнды (16 бит)						Константы	Указатели	Уровень												
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						K	H (16#)	P	I	N	
n																						●		3/5	● <sup>1</sup>		
d	●	●	●	●	●	●																	● <sup>2</sup>				

<sup>1</sup> Функция индексации имеется только для процессоров AnA, AnAS и AnU.

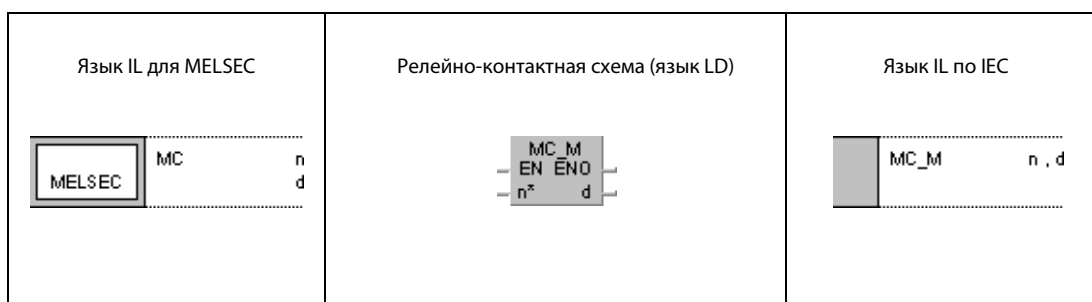
<sup>2</sup> Для команды MC расходуются 5 шагов, а для команды MCR – 3 шага. Более подробную информацию о количестве шагов в случае применения процессоров AnA, AnAS и AnU можно найти в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

**Операнды MELSEC Q**

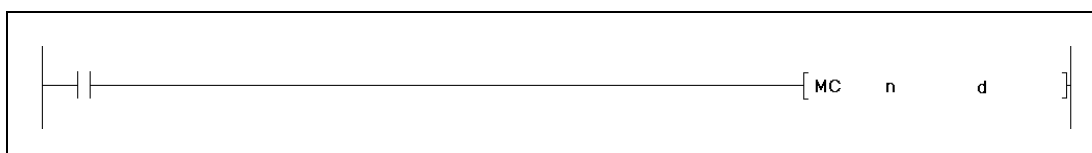
	Операнды										Флаг ошибки	Шагов
	Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы K, H (16#)	Иные			
	битовые	словные		битовые	словные				N	DY		
n	—	—	—	—	—	—	—	—	●		—	1/2
d	●	●	●	●	●	●	—	—		●	—	● <sup>1</sup>

<sup>1</sup> Для команды MC расходуются 2 шага, а для команды MCR – 1 шаг.

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
n	Уровень вложения (серия "A" = N0 – N7, серии "Q"/"System Q" = N0 – N14)	уровень
d	Операнд, включающий уровень вложения.	бит

**Принцип действия**

**Активация и деактивация отдельных областей программы**

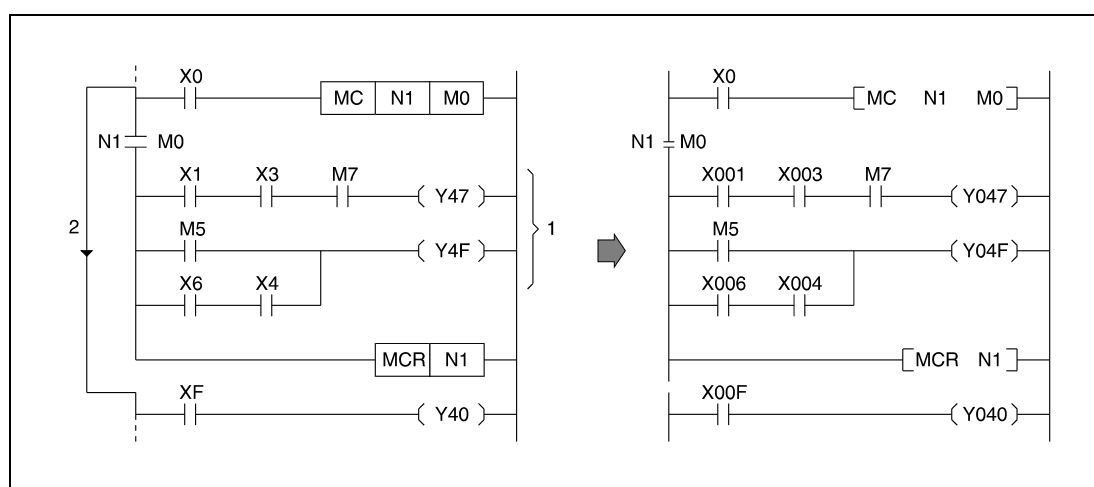
**Общие сведения**

Команда MC применяется для эффективного переключения между отдельными частями программы для выполнения отдельных процедур. После включения входного условия выполняется часть программы, находящаяся между целевым адресом d и командой MCR. Относящаяся к команде MC область программы устанавливается путем указания уровня (от N0 до N7 для серии "A" и от N0 до N14 для серий Q/System Q).

Так как в среде GX IEC Developer наглядное программирование команд MC/MCR не возможно, здесь для наглядности приведены релейно-контактные схемы среды GX Developer.

Принцип действия команды MC поясняет релейно-контактная схема. При выключенном входе X0 часть программы, расположенная на уровне 1 и обозначенная N1, пропускается (1). После включения X0 часть программы, расположенная на уровне N1, выполняется до команды MCR (2).

При программировании в режиме релейно-контактной схемы нет необходимости вставлять контакты команды MC в вертикальную шину. Они отображаются автоматически.



**MC Активация отдельных областей программы**

Команда MC является начальной организующей командой для вызова определенной части программы. Если входное условие команды MC включено, то адреса операндов, расположенные между командами MC и MCR, обрабатываются как обычно.

Расположенные между командами MC и MCR адреса операндов продолжают обрабатываться и после выключения входного условия команды MC. Время цикла программы из-за этого не укорачивается. Как только входное условие перестает выполняться, расположенные между командами MC и MCR операнды обрабатываются следующим образом:

Операнд	Обработка
10-миллисекундный таймер 100-миллисекундный таймер	Значение счета устанавливается на 0. Входной и выходной контакт выключаются
Фиксируемый 10-миллисекундный таймер (только серии "Q" и "System Q") Фиксируемый 100-миллисекундный таймер Счетчик	Значение счета и состояние входных контактов остаются прежними. Выходной контакт выключается.
Операнды в команде OUT	Все выходы выключаются.
Операнды в командах SET, RST и SFT	Фактическое состояние остается прежним.



**ПРИМЕЧАНИЕ** Если в части программы между командами MC и MCR имеется команда, для которой не требуется программировать непосредственное входное условие (например, FOR...NEXT, EI, DI и т. п.), то контроллер выполняет эту команду без учета входного условия команды MC.

В команде MC допускаются одинаковые номера уровней n – при условии, что устанавливаются различные адреса операндов d.

После установки команды MC включается контакт операнда, указанного в d. Если этот операнд запрограммирован в качестве входного условия и в другом месте программы, то контакт обрабатывается в качестве двойного контакта, т. е. контакты включаются и выключаются параллельно. Поэтому операнд, указанный в d, не следует использовать в других командах.

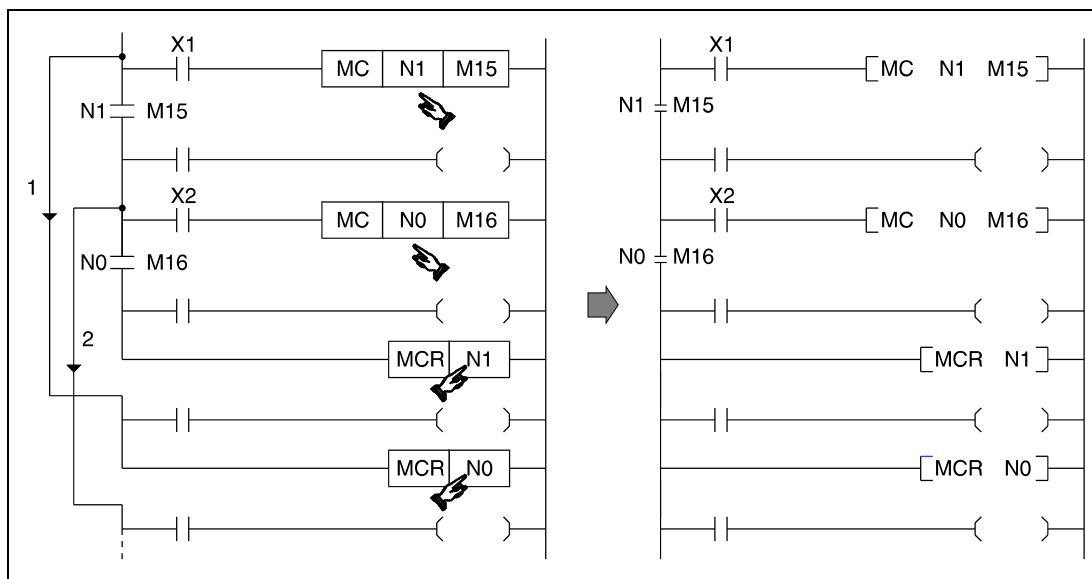
**MCR Дезактивация отдельных областей программы**

Команда MCR сбрасывает команду MC и обозначает конец организационной части программы.

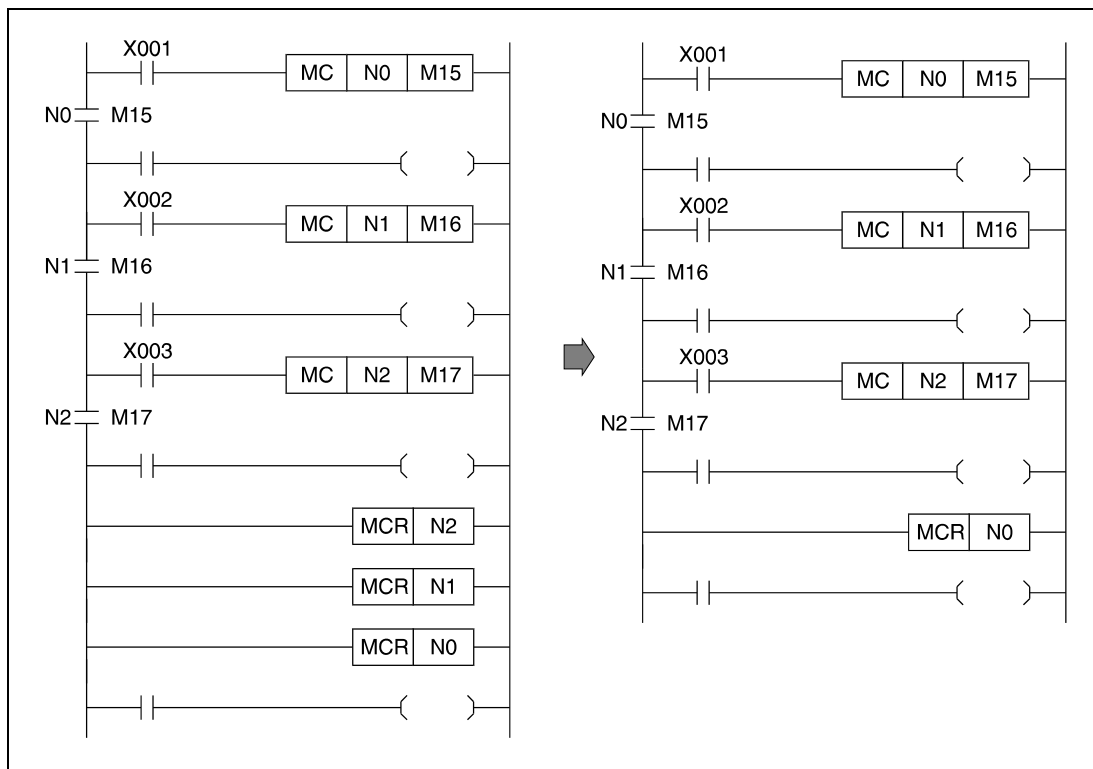
Команду MCR нельзя активировать через входной контакт.

При программировании адресов вложений необходимо соблюдать следующие пункты:

Вложения в серии "Q" или "System Q" могут иметь 15 уровней от N0 до N14, а в серии "A" – 8 уровней от N0 до N7. Первая область программы, вызываемая командой MC, должна иметь самый низкий адрес вложения, а первая команда MCR – самый высокий адрес вложения. Если адреса вложений присваиваются в иной последовательности, то выполняемые уровни (1, 2) сопоставляются неправильно и контроллер может неправильно выполнить программу. Это обстоятельство пояснено на следующей иллюстрации.



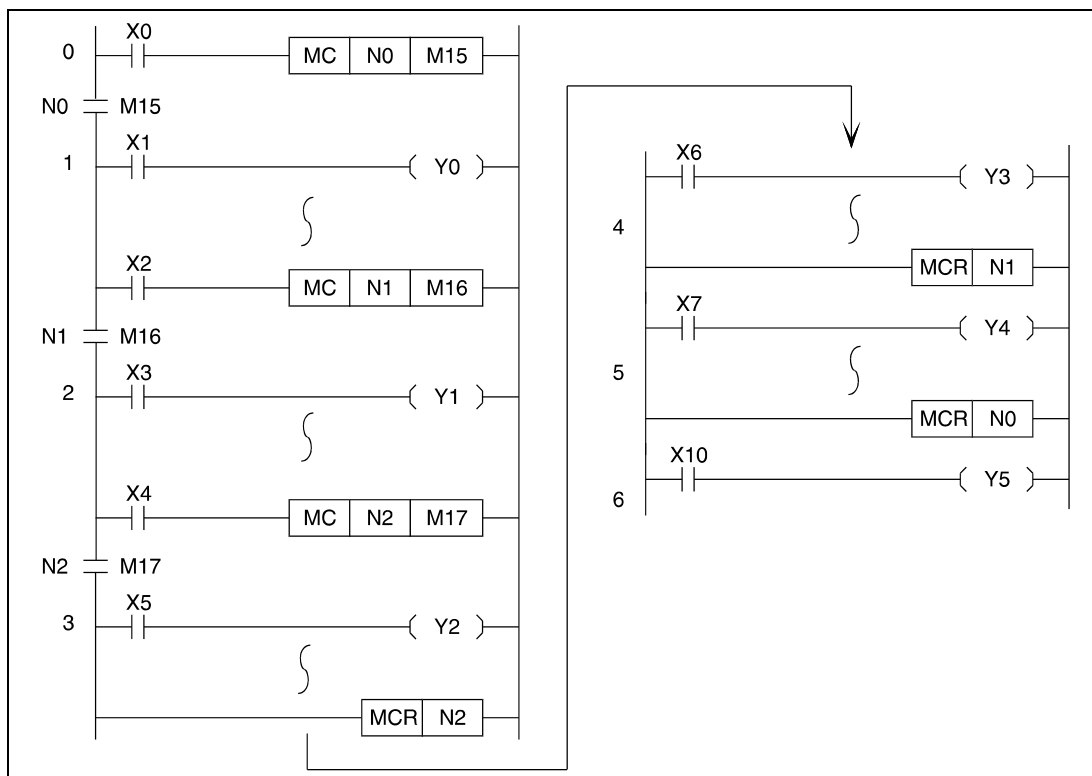
Если требуется запрограммировать несколько команд MCR подряд, то программу можно укоротить, запрограммировав для окончания всех частичных программ MC лишь одну команду MCR с самым низким адресом вложения.



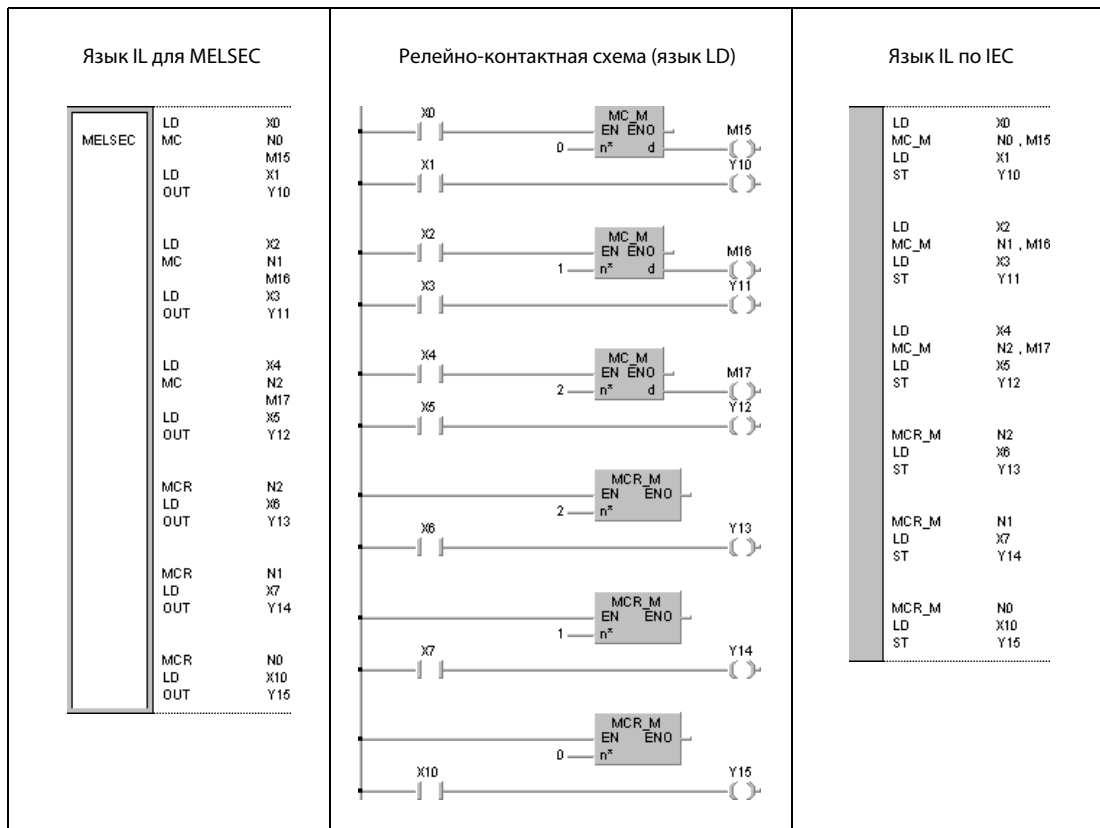
**Пример** MC, MCR

Для задания выполняемого уровня в команде MC программируется адрес вложения N. Для процессоров серий "Q" и "System Q" адреса вложений могут находиться в диапазоне между N0 и N14, а для процессоров серии "A" – между N0 и N7.

С помощью адресов вложений можно устанавливать последовательность выполнения областей MC программы. В этом примере показан пример программы с несколькими уровнями выполнения и использованием адресов вложений. Для лучшего понимания здесь также изображена релейно-контактная схема GX Developer.



Вышеприведенный пример дополнительно проиллюстрирован различными языками GX Developer (см. ниже).



## 5.6 Указание конца программы

### 5.6.1 FEND

**ПРИМЕЧАНИЕ** Эту команду не следует использовать в редакторах по стандарту IEC.

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

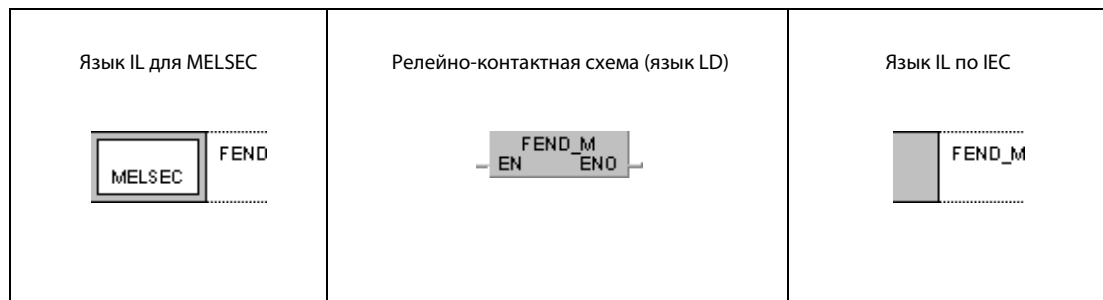
**Операнды MELSEC A**

Операнды																	Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011			
Битовые операнды							Словные операнды (16 бит)						Константы		Указатели	Уровень								
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K						N (16#)	P	I
																						1		●

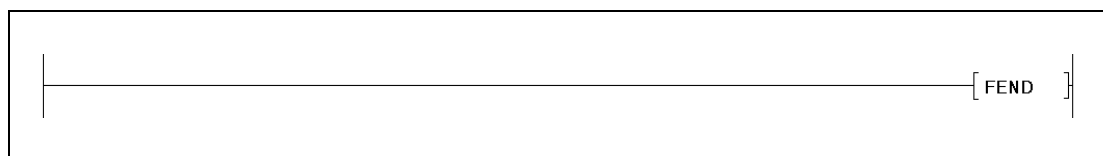
**Операнды MELSEC Q**

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непос. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы	Иные			
битовые	словные		битовые	словные							
—	—	—	—	—	—	—	—	—	—	SMO	1

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
—	—	—

**Принцип действия****Конец области программы****FEND Команда, заканчивающая область программы**

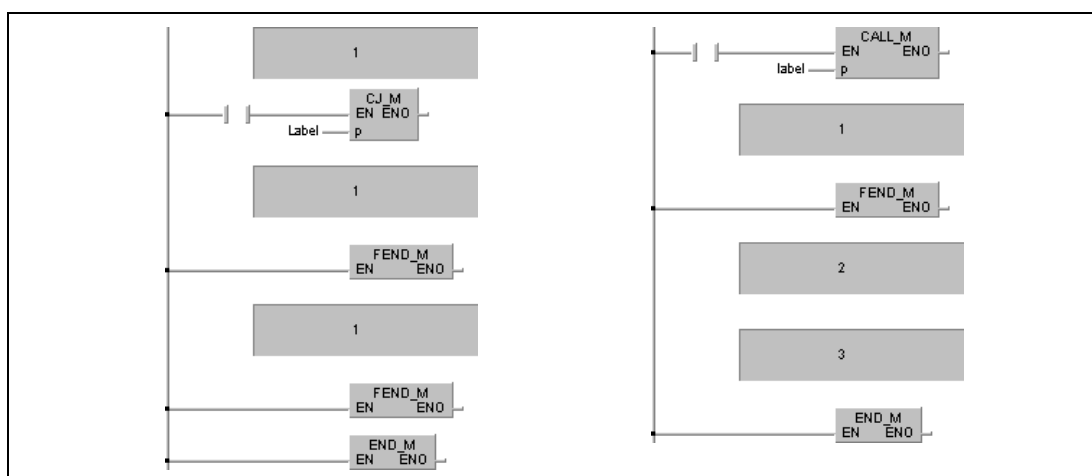
Команда FEND обозначает конец области программы. Это может быть область как основной программы, так и подпрограммы.

После выполнения команды FEND программа переходит к команде END. Выполнение внутренних процессов (обработка таймеров и счетчиков, самодиагностика центрального процессора) снова начинается с шага 0 программы.

В левом примере показано завершение областей программы при разветвлении на подпрограмму с помощью команды CJ.

При выполнении команды CJ программа выполняется до ближайшей команды FEND за той частью программы, в которую был выполнен переход. Если команда CJ не выполняется, то после достижения первой команды FEND программа переходит обратно к шагу 0.

В правом примере показано применение команды FEND для отделения области основной программы от области подпрограммы или программы прерывания.



1 основная программа

2 подпрограмма

3 подпрограмма прерывания

**ПРИМЕЧАНИЯ**

В среде GX Developer команду FEND должен вставлять пользователь. После отработки этого программного компонента (POU) никакие другие компоненты более не выполняются, так как они находятся за командой FEND.

Вместо программирования в этой среде можно программировать в стандартном редакторе МЭК (IEC). В этом случае среда GX IEC Developer вставляет команду FEND автоматически.

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- После выполнения команды CALL, FCALL, ECALL или EFCALL и перед обработкой команды RET выполняется команда FEND.  
(серия "Q"/"System Q" = код ошибки 4211).
- После выполнения команды FOR и перед обработкой команды NEXT выполняется команда FEND (серия "Q" = код ошибки 4200).
- Команда FEND выполняется во время программы прерывания и перед командой IRET (серия "Q"/"System Q" = код ошибки 4221).
- Команда FEND выполняется между командами CHKCLR и CHKEND (серия "Q"/"System Q" = код ошибки 4230).
- Команда FEND выполняется между командами IX и IXEND (серия "Q"/"System Q" = код ошибки 4231).

5.6.2 END

ПРИМЕЧАНИЕ Эту команду не следует использовать в редакторах по стандарту IEC.

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

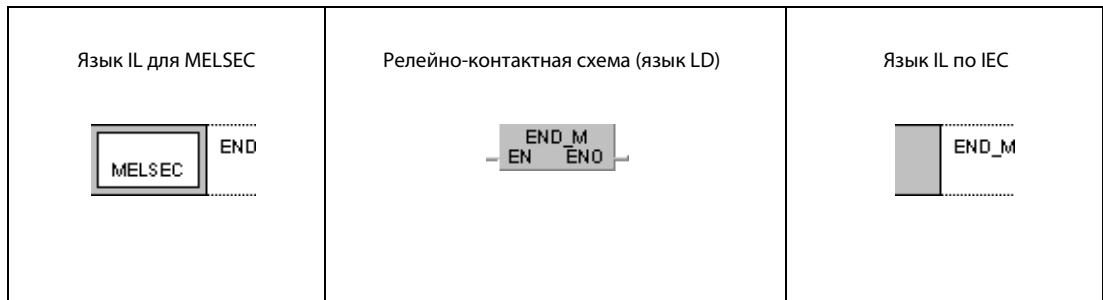
Операнды MELSEC A

Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011						
Битовые операнды						Словные операнды (16 бит)						Константы		Указатели							Уровень					
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						K	H (16#)	P	I	N	
																							1			●

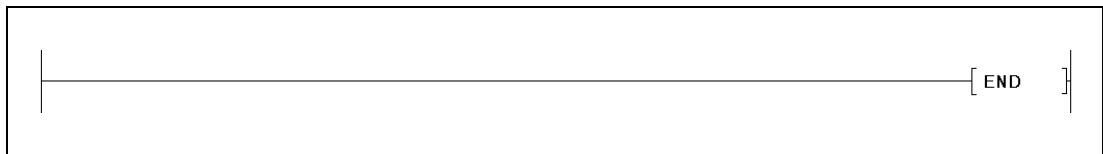
Операнды MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные				U			
—	—	—	—	—	—	—	—	—	—	SM0	1

GX IEC Developer



GX Developer



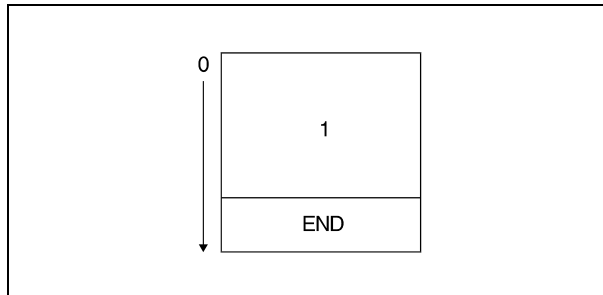
Переменные

Операнд	Значение	Тип данных
—	—	—



**Принцип действия****Конец основной программы или подпрограммы****END Команда, заканчивающая программу**

Команда END обозначает конец программы. На этом шаге программный цикл снова начинается с шага 0.

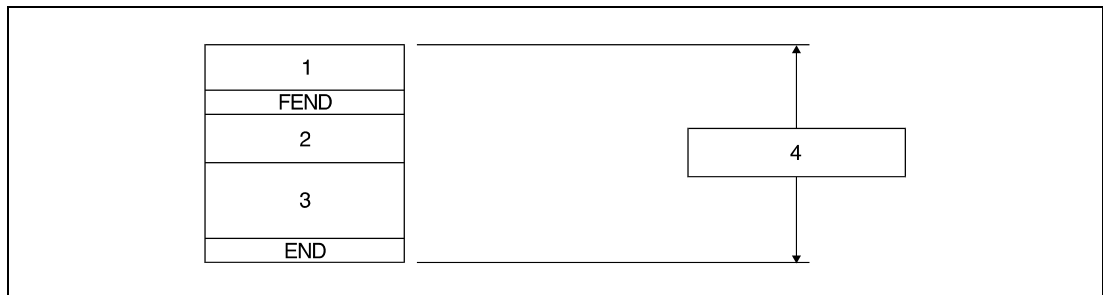


<sup>1</sup> подпрограмма

Команду END нельзя применять среди частей программы. Для завершения части программы следует применять команду FEND.

Если в программе не имеется команды END, то при запуске программы выработывается сообщение об ошибке и обработка программы контроллером прерывается. При отсутствии команды END сообщение об ошибке выработывается даже в том случае, если объем подпрограммы задан с помощью параметра.

Правильное использование команд END и FEND в отдельных частях программы пояснено на следующей иллюстрации.



<sup>1</sup> основная программа

<sup>2</sup> подпрограмма

<sup>3</sup> программа прерывания

<sup>4</sup> вся программа

**ПРИМЕЧАНИЕ** GX Developer и GX IEC Developer вставляют команду END автоматически.

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Цель перехода команды CJ, SCJ или JMP находится ниже команды END.
- Вызвана подпрограмма или подпрограмма прерывания, расположенная ниже команды END.
- После выполнения команды CALL, FCALL, ECALL или EFCALL и перед обработкой команды RET выполняется команда END.  
(серия "Q"/"System Q" = код ошибки 4211).
- После выполнения команды FOR и перед обработкой команды NEXT выполняется команда END (серия "Q"/"System Q" = код ошибки 4200).
- Команда END выполняется во время программы прерывания и перед командой IRET (серия "Q"/"System Q" = код ошибки 4221).
- Команда END выполняется между командами CHK CIR и CHK END (серия "Q"/"System Q" = код ошибки 4230).
- Команда END выполняется между командами IX и IXEND (серия "Q"/"System Q" = код ошибки 4231).

## 5.7 Прочие команды

### 5.7.1 STOP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011			
Битовые операнды				Словные операнды (16 бит)						Константы		Указатели	Уровень										
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						K	H (16#)	P
																					1		

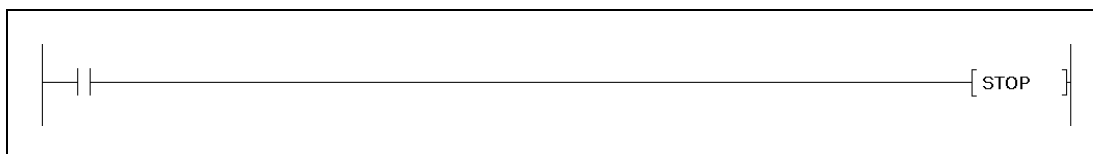
Операнды MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные				U			
—	—	—	—	—	—	—	—	—	—	SMO	1

GX IEC Developer



GX Developer



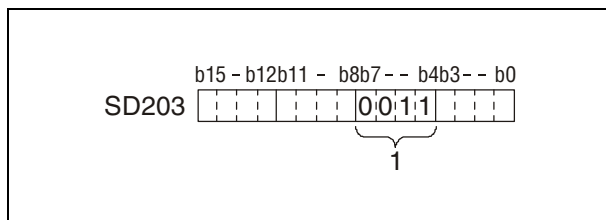
Переменные

Операнд	Значение	Тип данных
—	—	—

**Принцип действия****Прерывание обработки****STOP Команда прерывания обработки**

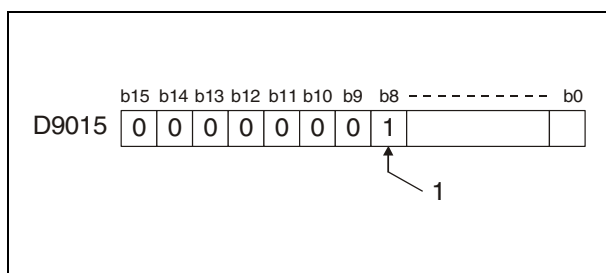
После включения входного условия все выходы (Y) выключаются и отработка программы контроллера останавливается. Эта функция соответствует переключению выключателя "RUN/STOP" на модуле центрального процессора в положение "STOP".

При выполнении команды STOP в процессорах серий "Q" и "System Q" биты с 5-го по 8-й (от b4 до b7) регистра SD203 устанавливаются на значение 3.



<sup>1</sup> число 3 в двоичном формате

При выполнении команды STOP процессором MELSEC серии "A" девятый бит (b8) специального регистра D9015 устанавливается на 1.



<sup>1</sup> Устанавливается на 1.

Чтобы после выполнения команды STOP возобновить работу контроллера, следует на короткое время переключить выключатель "RUN/STOP" с "RUN" на "STOP", а затем снова на "RUN".

Если после выполнения команды STOP выключатель "RESET" переключен на "LATCH CLEAR", это никак не проявляется на содержимом промежуточной памяти. Для стирания содержимого промежуточной памяти следует сначала установить выключатель "RUN/STOP" на "STOP", а затем установить выключатель "RESET" на "L.CL." ("LATCH CLEAR").

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- После выполнения команды CALL, FCALL, ECALL или EFCALL и перед обработкой команды RET выполняется команда END.  
(серия "Q"/"System Q" = код ошибки 4211)
- После выполнения команды FOR и перед обработкой команды NEXT выполняется команда END (серия "Q"/"System Q" = код ошибки 4200).
- Команда END выполняется во время программы прерывания и перед командой IRET (серия "Q"/"System Q" = код ошибки 4221).
- Команда END выполняется между командами CHKCIR и CHKEND (серия "Q"/"System Q" = код ошибки 4230).
- Команда END выполняется между командами IX и IXEND (серия "Q"/"System Q" = код ошибки 4231).

**Пример**

STOP

В следующей программе при включении X8 обработка останавливается. Все последующие шаги программы выполняются после переключения выключателя "RUN/STOP" с "RUN" на "STOP", а затем снова на "RUN".



5.7.2 NOP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●


Операнды MELSEC A

Операнды																	Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки			
Битовые операнды							Словные операнды (16 бит)						Константы		Указатели							Уровень		
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K				H (16#)	P	I	N	M9012
																						1		

Операнды MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□\G□□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные							
—	—	—	—	—	—	—	—	—	—	—	1

GX IEC Developer

Язык IL для MELSEC  	Релейно-контактная схема (язык LD)	Язык IL по IEC
---	------------------------------------	----------------

GX Developer

--

Переменные

Операнд	Значение	Тип данных
—	—	—

**Принцип действия**

**Пустые шаги**  
**NOP Пустой шаг в программе**

Команда NOP (от слов "No OPeration" – "никакой операции") обозначает пустой шаг в программе и никак не сказывается на обработке имеющихся частей программы. С помощью этой команды в программе, написание которой еще не завершено, вырабатывается логический пустой шаг, который позднее можно заполнить другими командами.

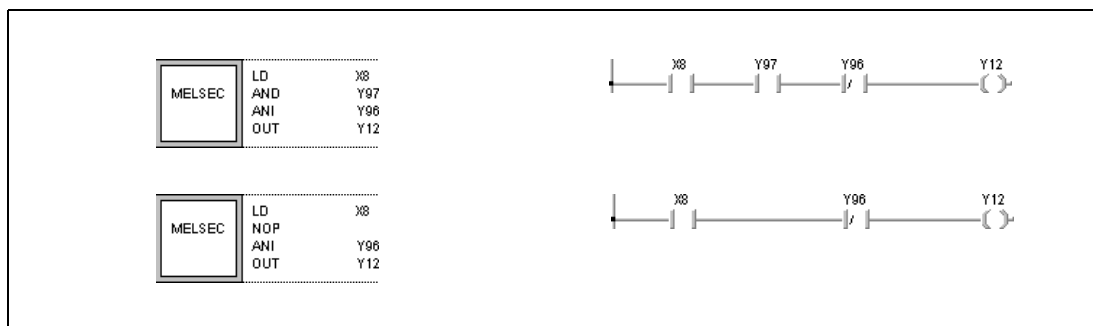
Вставлять команду NOP целесообразно в следующих случаях:

- чтобы зарезервировать место для отладки программы;
- чтобы стереть команду (перезаписав ее командой NOP), не изменив нумерацию шагов программы;
- для временного стирания команды с последующим редактированием.

**ПРИМЕЧАНИЕ** По окончании работы над какой-либо частью программы команды NOP следует по возможности стирать, чтобы сократить время цикла программы.

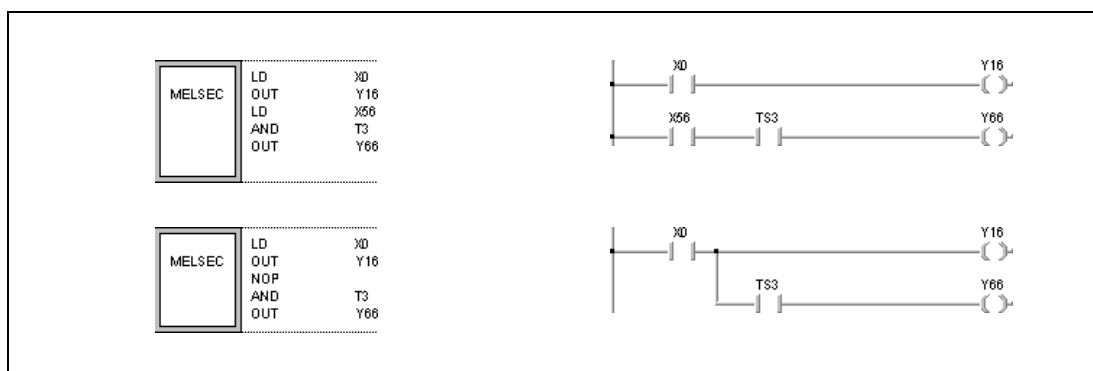
**Пример 1** NOP

В следующем примере для отладки программы соединения (AND) заменены пустыми шагами.



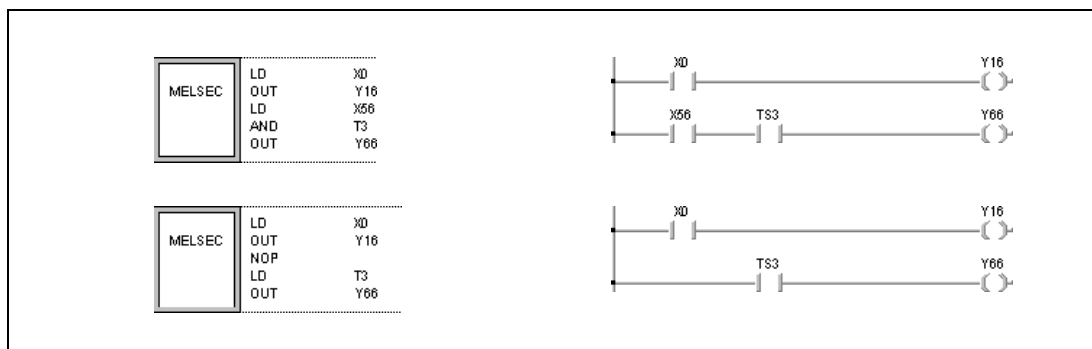
**Пример 2** NOP

В следующей программе команда LD заменена командой NOP.



**Пример 3** NOP

В следующей программе команда LD заменена командой NOP.



**ПРИМЕЧАНИЕ** *Заменять командой NOP входной контакт (LD, LDI) следует очень осторожно, так как это сильно влияет на всю программную логику.*



## 6 Прикладные команды, часть I

Прикладные команды, часть I, включают в себя команды, оперирующие числовыми 16-битными и 32-битными данными, числами с плавающей запятой и строковыми величинами. Главным образом, эти команды представляют собой операции сравнения и вычислительные операции.

Вид команды	Значение
Операторы сравнения	Сравнение данных, например, =, >, ≥ и т. п.
Арифметические операции	Сложение, вычитание, умножение и деление двоичных и двоично-десятичных данных, чисел с плавающей запятой и двоичных блоков данных, соединение строковых величин, приращение, декремент
Команды конвертирования	Конвертирование данных, например, BCD → BIN или BIN → BCD
Команды переноса	Перенос, замена и инвертирование данных
Команды разветвления программы	Переходы, вызов подпрограммы
Команды для вызова программ прерывания	Вызов программ прерывания
Команды обновления данных	Обновление канала связи и интерфейса ввода-вывода
Прочие команды	Однофазные и двухфазные возрастающие и убывающие счетчики, программируемые таймеры, таймер особой функции, команда позиционирования, линейно изменяющийся сигнал, счетчик импульсов, импульсный выход, широтно-импульсная модуляция, матрица ввода

## 6.1 Операторы сравнения

Операторы сравнения могут сравнивать два блока данных (например, равно =, больше >, меньше < и т. п.). Операторы сравнения программируются аналогично тому, как это описано для соответствующих команд из базового набора команд:

LD, LDI ⇒ LD =, LDD =

AND, ANI ⇒ AND =, ANDD =

OR, ORI ⇒ OR =, ORD =

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC	Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC
= равно	LD =	LD_EQ_M	≤ меньше или равно	LD <=	LD_LE_M
	AND =	AND_EQ_M		AND <=	AND_LE_M
	OR =	OR_EQ_M		OR <=	OR_LE_M
	LDD =	LDD_EQ_M		LDD <=	LDD_LE_M
	ANDD =	ANDD_EQ_M		ANDD <=	ANDD_LE_M
	ORD =	ORD_EQ_M		ORD <=	ORD_LE_M
	LDE =	LD_EEQ_M		LDE <=	LD_ELE_M
	ANDE =	AND_EEQ_M		ANDE <=	AND_ELE_M
	ORE =	OR_EEQ_M		ORE <=	OR_ELE_M
	LD\$ =	LD_STRING_EQ_M		LD\$ <=	LD_STRING_LE_M
	AND\$ =	AND_STRING_EQ_M		AND\$ <=	AND_STRING_LE_M
	OR\$ =	OR_STRING_EQ_M		OR\$ <=	OR_STRING_LE_M
	BKCOMP =	BKCOMP_EQ_M		BKCOMP <=	BKCOMP_LE_M
BKCOMP = P	BKCOMP_EQP_M	BKCOMP <=P	BKCOMP_LEP_M		
≠ неравно	LD <>	LD_NE_M	< меньше	LD <	LD_LT_M
	AND <>	AND_NE_M		AND <	AND_LT_M
	OR <>	OR_NE_M		OR <	OR_LT_M
	LDD <>	LDD_NE_M		LDD <	LDD_LT_M
	ANDD <>	ANDD_NE_M		ANDD <	ANDD_LT_M
	ORD <>	ORD_NE_M		ORD <	ORD_LT_M
	LDE <>	LD_ENE_M		LDE <	LD_ELT_M
	ANDE <>	AND_ENE_M		ANDE <	AND_ELT_M
	ORE <>	OR_ENE_M		ORE <	OR_ELT_M
	LD\$ <>	LD_STRING_NE_M		LD\$ <	LD_STRING_LT_M
	AND\$ <>	AND_STRING_NE_M		AND\$ <	AND_STRING_LT_M
	OR\$ <>	OR_STRING_NE_M		OR\$ <	OR_STRING_LT_M
	BKCOMP <>	BKCOMP_NE_M		BKCOMP <	BKCOMP_LT_M
BKCOMP <> P	BKCOMP_NEP_M	BKCOMP < P	BKCOMP_LTP_M		

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC	Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC
> больше	LD >	LD_GT_M	≥ больше или равно	LD >=	LD_GE_M
	AND >	AND_GT_M		AND >=	AND_GE_M
	OR >	OR_GT_M		OR >=	OR_GE_M
	LDD >	LDD_GT_M		LDD >=	LDD_GE_M
	ANDD >	ANDD_GT_M		ANDD >=	ANDD_GE_M
	ORD >	ORD_GT_M		ORD >=	ORD_GE_M
	LDE >	LD_EGT_M		LDE >=	LD_EGE_M
	ANDE >	AND_EGT_M		ANDE >=	AND_EGE_M
	ORE >	OR_EGT_M		ORE >=	OR_EGE_M
	LD\$ >	LD_STRING_GT_M		LD\$ >=	LD_STRING_GE_M
	AND\$ >	AND_STRING_GT_M		AND\$ >=	AND_STRING_GE_M
	OR\$ >	OR_STRING_GT_M		OR\$ >=	OR_STRING_GE_M
	BKCMPP >	BKCMPP_GT_M		BKCMPP >=	BKCMPP_GE_M
BKCMPP >P	BKCMPP_GTP_M	BKCMPP >=P	BKCMPP_GEP_M		

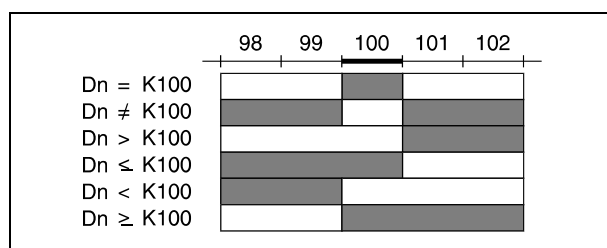
**ПРИМЕЧАНИЕ** В редакторах IEC следует использовать команды по стандарту IEC.

**Команды по стандарту IEC**

Функция	Команда IEC	Значение
=	EQ	Equal
<>	NE	Not Equal
<=	LE	Less Equal
<	LT	Less Than
>=	GE	Greater Equal
>	GT	Greater Than

**Условия выполнения**

На рисунке ниже пояснены условия выполнения различных операторов сравнения.

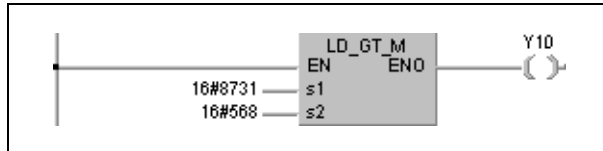


■ = 1 = включено  
□ = 0 = отключено

**ПРИМЕЧАНИЯ** Все данные, участвующие в операторах сравнения, обрабатываются в качестве двоичных значений.

При сравнении  $16\#8000 > 16\#7999$  результат сравнения устанавливается на ЛОЖНО, хотя следовало бы ожидать ИСТИННО. Причина заключается в том, что значения преобразуются в двоичный вид и тем самым устанавливается также бит 15 (b15). Если бит 15 установлен, число становится отрицательным.

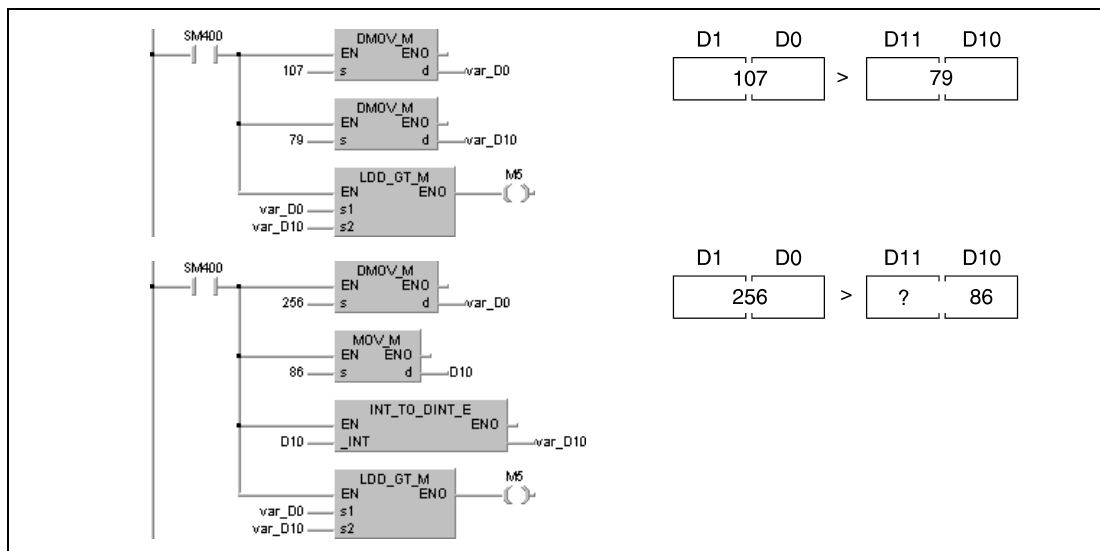
**Пример 1** Сравнение 2 четырехзначных двоично-десятичных значений



8731H интерпретируется как -30927, а 568H – как 1384. Так как сравнение этих чисел дает  $-30927 > 1384$ , выход Y10 не устанавливается.

Для функций сравнения 32-битных данных входное числовое значение следует определять с помощью 32-битной команды, например, DMOV. Если для определения используется 16-битная команда (например, MOV), функция может работать неправильно, так как при 32-битном сравнении всегда используется значение данных из n и (n + 1).

**Пример 2** Функция сравнения 32-битных данных



В этом примере показаны две операции сравнения 32-битных данных. В верхней программе M5 включается, так как здесь оба значения были определены с помощью 32-битной команды DMOV.

Результат нижней программы невозможно предугадать, так как значение старших байтов не определено однозначно.

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

6.1.1 =, <, >, >, <=, <, >=

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

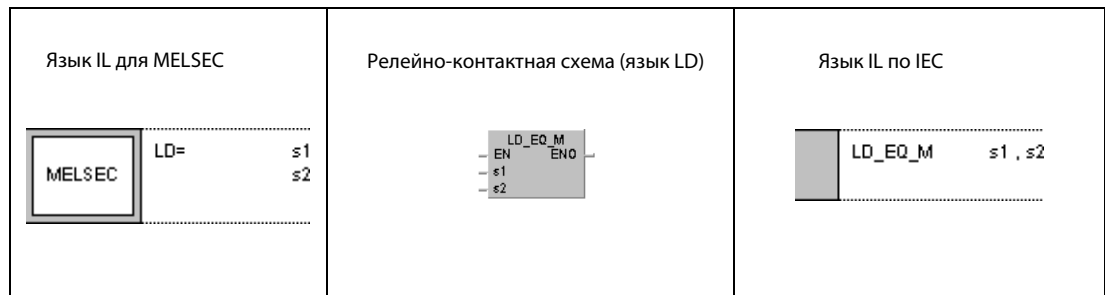
		Операнды															Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011					
		битовые						словные (16 бит)						Константы		Указатели						Уровень				
		X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V	K	H (16#)	P	I
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					K1 ↓ K4	5/7	●		●
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●									

<sup>1</sup> Если выполняется функция индексации, длина блока битового операнда не равна K4 или головной адрес битового операнда не кратен 8 (или 16 в случае процессоров A3H, A3M, AnA, AnAS или AnU), то расходуется 7 шагов. Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

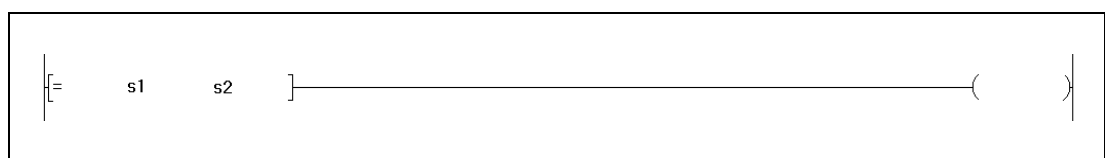
Операнды MELSEC Q

		Операнды								Флаг ошибки	Шагов	
		Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
		битовые	словные		битовые	словные						
s1	●	●	●	●	●	●	●	●	—	—	3	
s2	●	●	●	●	●	●	●	●	—	—		

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
s1	Сравниваемые данные или операнд, в котором сохранены сравниваемые данные.	BIN, 16 бит
s2		

**Принцип действия**

**Сравнение 16-битных данных**

**Операторы сравнения =, <>, >, <=, <, >=**

16-битная команда сравнения состоит из самой команды и данных s1 и s2, которые требуется сравнить друг с другом.

Операторы сравнения, представленные в графическом виде, обрабатываются в качестве замыкающего контакта. Сравниваются 16-битные данные.

Следующий выход переключается в зависимости от результата функции сравнения. Соответствующие условия включения названы в следующей таблице.

Символ команды	Состояние выхода	
	1, если:	0, если:
=	s1 = s2	s1 ≠ s2
<>	s1 ≠ s2	s1 = s2
>	s1 > s2	s1 ≤ s2
<=	s1 ≤ s2	s1 > s2
<	s1 < s2	s1 ≥ s2
>=	s1 ≥ s2	s1 < s2

**ПРИМЕЧАНИЯ**

Все данные, участвующие в операторах сравнения, обрабатываются в качестве двоичных значений.

При сравнении 16#8000 > 16#7999 результат сравнения устанавливается на ЛОЖНО, хотя следовало бы ожидать ИСТИННО. Причина этого заключается в том, что значения преобразуются в двоичный вид и тем самым устанавливается также бит 15 (b15). Если бит 15 установлен, число становится отрицательным.

**Пример 1**

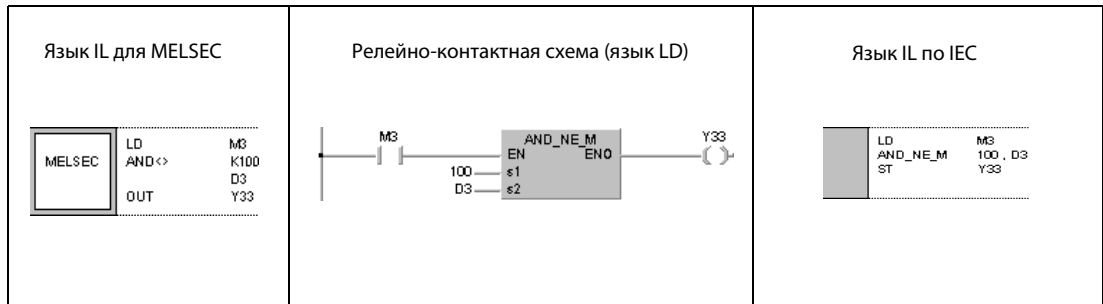
Команда сравнения =

Следующая программа сравнивает данные с X0 по XF со значением данных в D3. Если эти значения равны, устанавливается Y33.



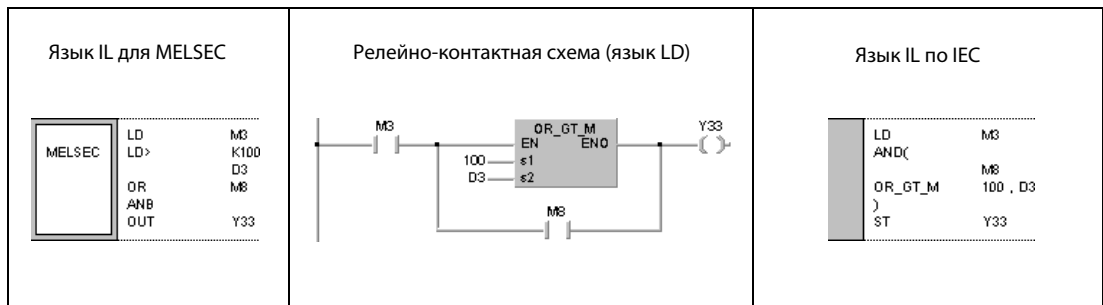
**Пример 2** Команда сравнения <>

Следующая программа сравнивает двоичное значение 100 со значением данных в D3. Если значение в D3 не равно 100 и M3 установлен, устанавливается Y33.



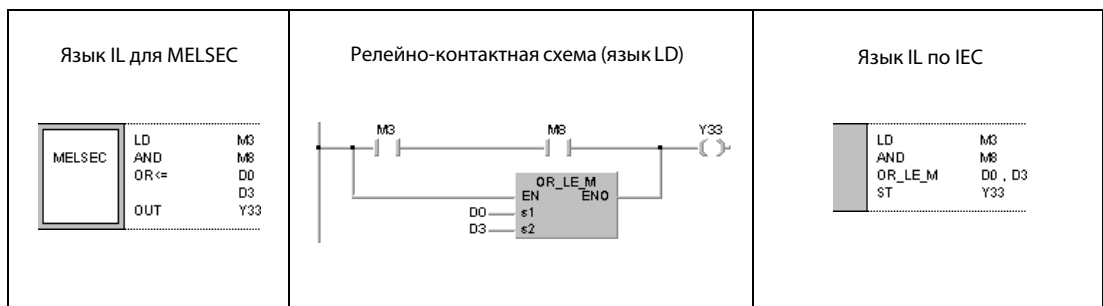
**Пример 3** Команда сравнения >

Следующая программа сравнивает двоичное значение 100 со значением данных в D3. Если значение в D3 меньше 100 и M3 установлен, устанавливается Y33. Если M8 и M3 установлены, то устанавливается и Y33.



**Пример 4** Команда сравнения <=

Следующая программа сравнивает значение в D0 со значением в D3. Если значение в D0 меньше или равно D3, устанавливается Y33. Если M8 и M3 установлены, то устанавливается и Y33.



6.1.2 D =, D <>, D >, D <=, D <, D >=

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

		Операнды														Указа-тели	Уро-вень	Длина блока	Шагов	Индекс	Флаг пере-носа	Флаг ошиб-ки				
		битовые						словные (16 бит)						Константы												
		X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1								Z	V	K	H (16#)
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				К1	11	●		●
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				К8	1			

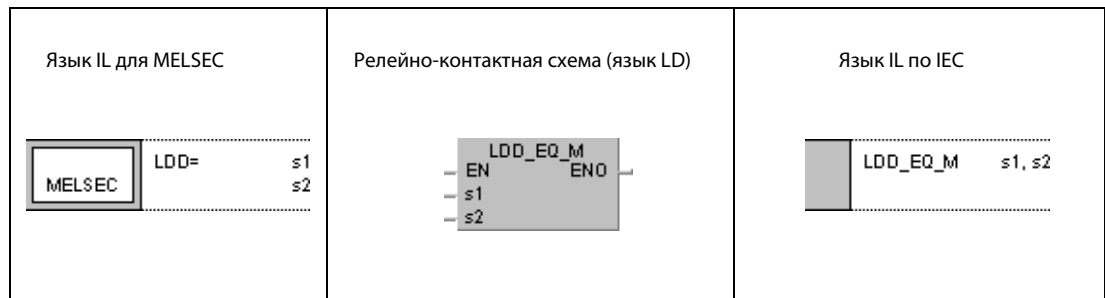
<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

Операнды MELSEC Q

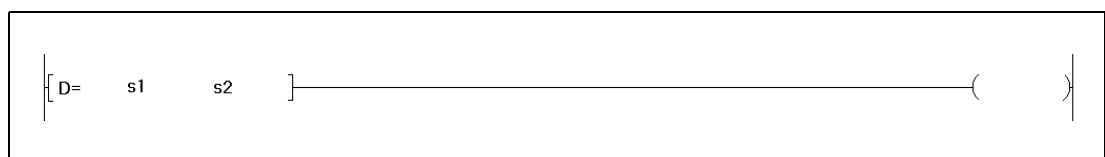
		Операнды							Флаг ошибки	Шагов		
		Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn			Константы K, H (16#)	Иные
		битовые	словные		битовые	словные						
s1	●	●	●	●	●	●	●	●	●	—	—	3 <sup>1)</sup>
s2	●	●	●	●	●	●	●	●	●	—	—	

<sup>1</sup> Количество шагов программы зависит от типа центрального процессора и используемых операндов.  
 При использовании процессоров серии QnA или однопроцессорного модуля ЦП серии "Q": 3  
 При использовании многопроцессорного модуля ЦП серии "Q" и внутренних словных операндов (кроме регистра файлов ZR): 5  
 Константы: 5  
 Битовые операнды с адресом, кратным 16, имеющие обозначение битового блока K8 и не индексируемые: 5  
 При использовании многопроцессорного модуля ЦП серии "Q" и иных операндов кроме вышеуказанных: 3

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
s1	Сравниваемые данные или операнд, в котором сохранены сравниваемые данные.	BIN, 32 бита
s2		



**Принцип действия**

**Сравнение 32-битных данных**

**Операторы сравнения D=, D<>, D>, D<=, D<, D>=**

32-битная команда сравнения состоит из самой команды и данных s1 и s2, которые требуется сравнить друг с другом.

Операторы сравнения, представленные в графическом виде, обрабатываются в качестве замыкающего контакта. Сравниваются 32-битные данные.

Следующий выход переключается в зависимости от результата функции сравнения. Соответствующие условия включения названы в следующей таблице.

Символ команды	Состояние выхода	
	1, если:	0, если:
D =	s1 = s2	s1 ≠ s2
D <>	s1 ≠ s2	s1 = s2
D >	s1 > s2	s1 ≤ s2
D <=	s1 ≤ s2	s1 > s2
D <	s1 < s2	s1 ≥ s2
D >=	s1 ≥ s2	s1 < s2

**ПРИМЕЧАНИЯ**

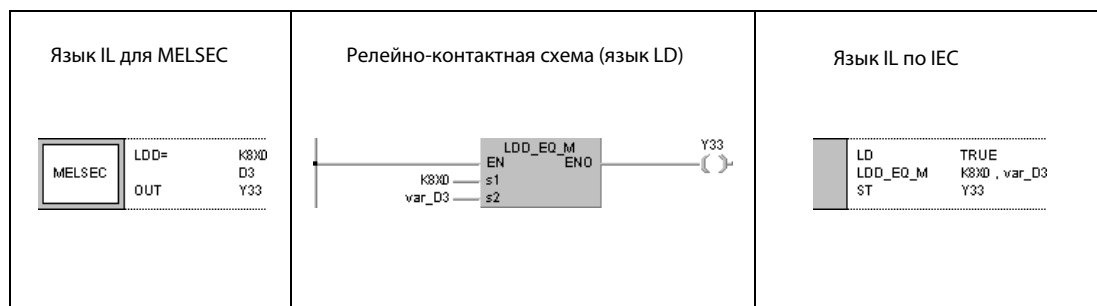
Все данные, участвующие в операторах сравнения, обрабатываются в качестве двоичных значений.

При сравнении 16#8000 > 16#7999 результат сравнения устанавливается на ЛОЖНО, хотя следовало бы ожидать ИСТИННО. Причина этого заключается в том, что значения преобразуются в двоичный вид и тем самым устанавливается также бит 15 (b15). Если бит 15 установлен, число становится отрицательным.

**Пример 1**

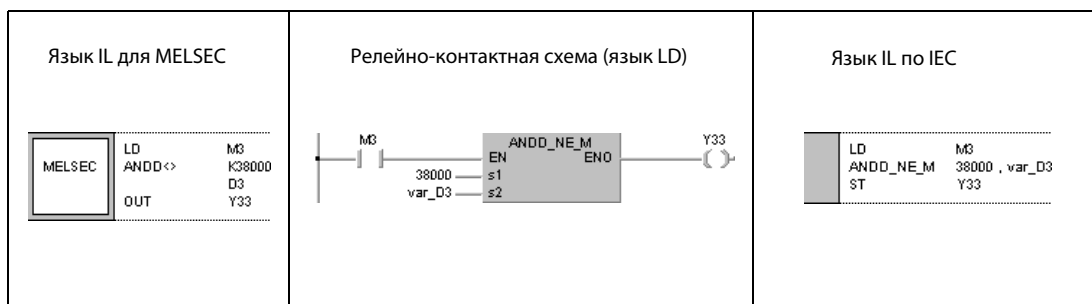
Команда сравнения D =

Следующая программа сравнивает данные с X0 по X1F со значением данных в D3 и D4. Если эти значения равны, устанавливается Y33.



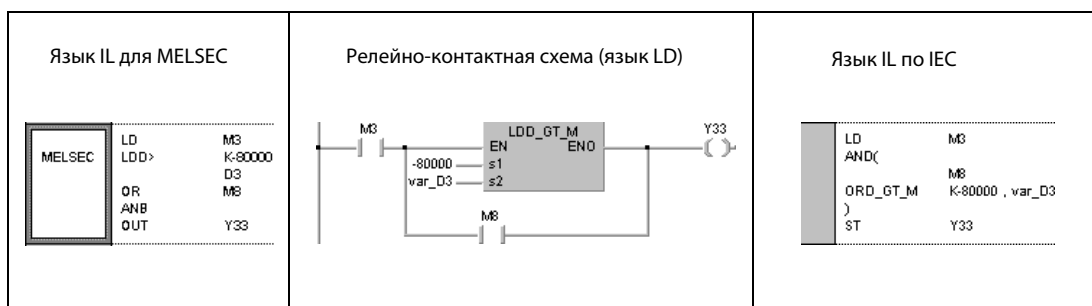
**Пример 2** Команда сравнения D <>

Следующая программа сравнивает двоичное значение 38000 со значением в D3 и D4. Если значение в D3 и D4 не равно 38000 и M3 установлен, устанавливается Y33.



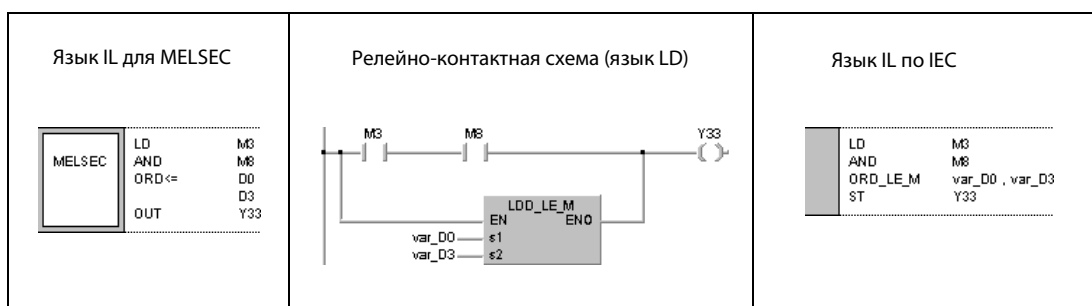
**Пример 3** Команда сравнения D >

Следующая программа сравнивает двоичное значение -80000 со значением в D3 и D4. Если значение в D3 и D4 меньше -80000 и M3 установлен, устанавливается Y33. Выход Y33 устанавливается также в том случае, если установлены M3 и M8.



**Пример 4** Команда сравнения D <=

Следующая программа сравнивает значение в D0 и D1 со значением в D3 и D4. Если значение в D3 и D4 больше или равно D0 и D1, устанавливается Y33. Выход Y33 устанавливается также в том случае, если установлены M3 и M8.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

**6.1.3 E =, E <>, E >, E <=, E <, E >=**

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

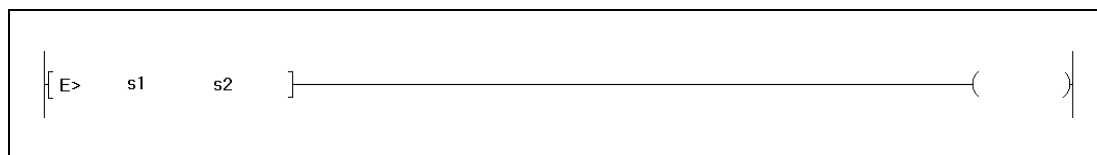
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы E			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	●	●	—	●	—	3	
s2	—	●	●	—	●	●	—	●	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s1	Сравниваемые данные или операнд, в котором сохранены сравниваемые данные.	REAL
s2		

**Принцип действия****Сравнение чисел с плавающей запятой****Операторы сравнения E =, E <>, E >, E <=, E <, E >=**

Команда сравнения для чисел с плавающей запятой состоит из самой команды и данных s1 и s2, которые требуется сравнить друг с другом.

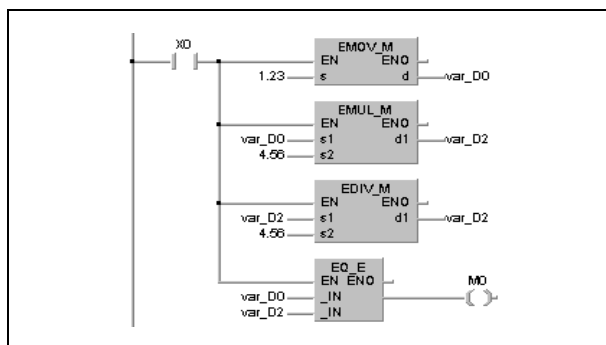
Операторы сравнения, представленные в графическом виде, обрабатываются в качестве замыкающего контакта. Сравниваются числа с плавающей запятой.

Следующий выход переключается в зависимости от результата функции сравнения. Соответствующие условия включения названы в следующей таблице.

Символ команды	Состояние выхода	
	1, если:	0, если:
E=	$s1 = s2$	$s1 \neq s2$
E<>	$s1 \neq s2$	$s1 = s2$
E>	$s1 > s2$	$s1 \leq s2$
E<=	$s1 \leq s2$	$s1 > s2$
E<	$s1 < s2$	$s1 \geq s2$
E>=	$s1 \geq s2$	$s1 < s2$

**ПРИМЕЧАНИЕ**

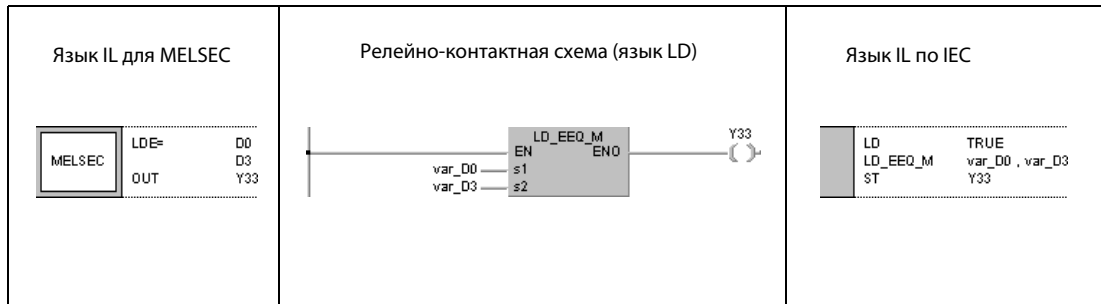
При сравнении чисел с плавающей запятой следует учитывать, что при некоторых обстоятельствах два значения, которые перед операцией были идентичны, после этой операции в результате ошибки округления уже не идентичны. В этом случае маркер M0 в нижнем примере не устанавливается.

**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию можно найти в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

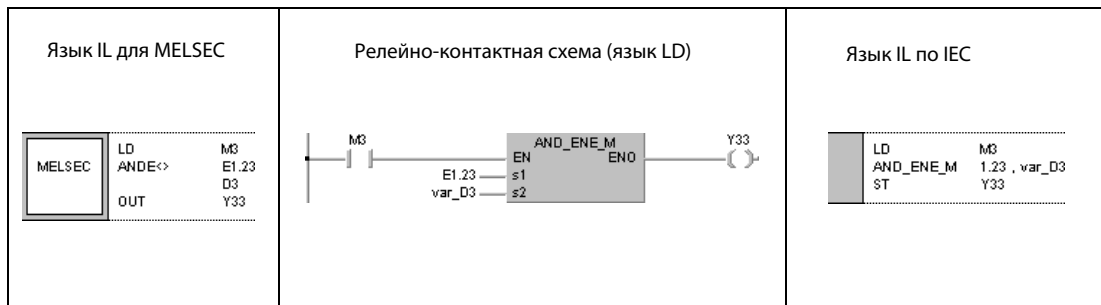
**Пример 1** Команда сравнения E =

Следующая программа сравнивает число с плавающей запятой, хранящееся в D0 и D1, с числом с плавающей запятой, хранящимся в D3 и D4. Если эти значения равны, устанавливается Y33.



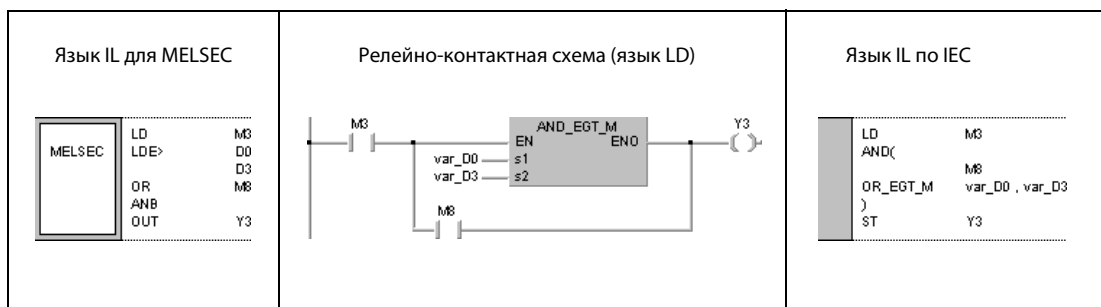
**Пример 2** Команда сравнения E <>

Следующая программа сравнивает число с плавающей запятой 1.23 с числом с плавающей запятой, хранящимся в D3 и D4. Если значение в D3 и D4 не равно 1.23 и маркер M3 установлен, устанавливается Y33.



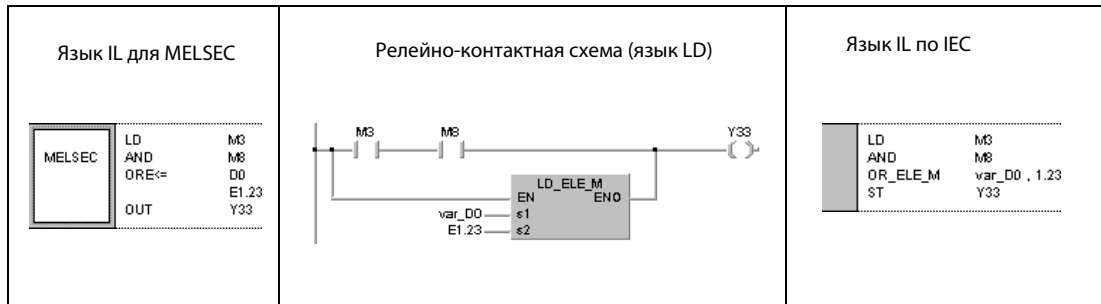
**Пример 3** Команда сравнения E >

Следующая программа сравнивает число с плавающей запятой, хранящееся в D0 и D1, с числом с плавающей запятой, хранящимся в D3 и D4. Если значение в D3 и D4 меньше значения в D0 и D1 и маркер M3 установлен, устанавливается Y3. Выход Y3 устанавливается также в том случае, если установлены маркеры M3 и M8.



**Пример 4** Команда сравнения E < =

Следующая программа сравнивает число с плавающей запятой, хранящееся в D0 и D1, с числом с плавающей запятой 1.23. Если значение в D0 и D1 меньше или равно 1.23, устанавливается Y33. Выход Y33 устанавливается также в том случае, если установлены M3 и M8.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

**6.1.4 \$ =, \$ < >, \$ >, \$ < =, \$ <, \$ > =**

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

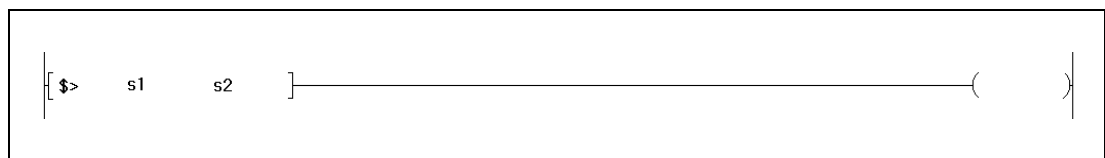
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы \$			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	●	—	SM0	3
s2	—	●	●	—	—	—	—	●	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s1	Первый адрес сравниваемых данных или операнд, в котором сохранены сравниваемые данные.	строковая величина
s2		

**Принцип действия**

**Сравнение строковых величин**

**Операторы сравнения \$ =, \$ <>, \$ >, \$ <=, \$ <, \$ >=**

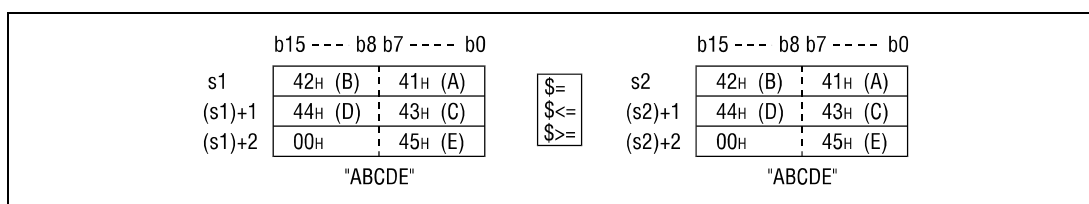
Команда для сравнения строковых величин состоит из самой команды и данных s1 и s2, которые требуется сравнить друг с другом.

Операторы сравнения, представленные в графическом виде, обрабатываются в качестве замыкающего контакта.

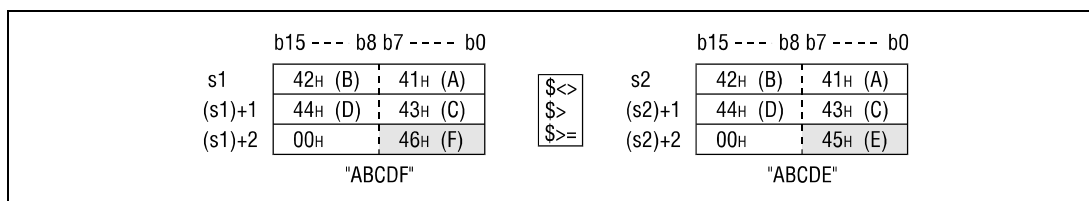
Сравнение строковых величин в кодировке ASCII происходит знак за знаком, начиная с первого знака строковой величины.

Хранящиеся в s1 и s2 строковые величины начинаются с самого младшего байта (1-й знак) и заканчиваются кодом "00H".

Если обе строковые величины имеют одинаковое содержимое и одинаковый размер, результат команд \$ =, \$ <= и \$ >= равен 1.

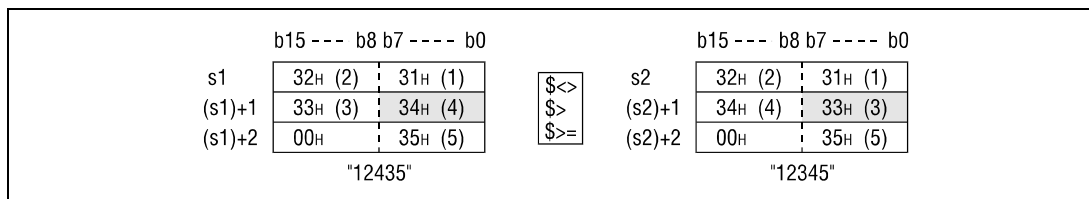


Если строковые величины различаются, то строковая величина с более высоким кодом знака считается большей.



В вышеприведенном случае команды \$ <>, \$ > и \$ >= дают результат 1.

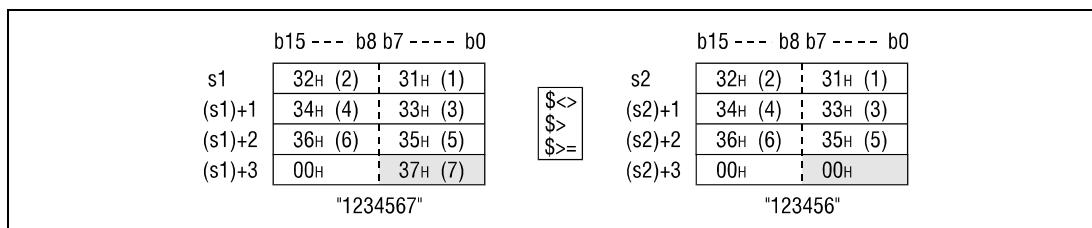
Если строковые величины различаются, то от числовой величины кода первого отличающегося знака зависит, какая из строковых величин считается большей.



В вышеприведенном случае команды \$ <>, \$ > и \$ >= дают результат 1.



Если строковые величины имеют различную длину, то более длинная строковая величина считается большей.



В вышеприведенном случае команды \$ < >, \$ > и \$ > = дают результат 1.

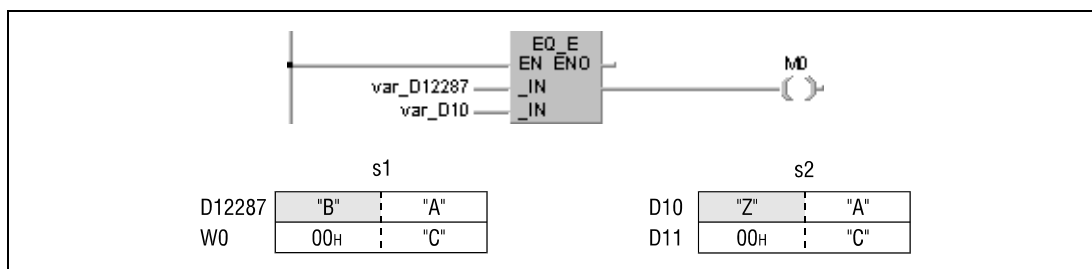
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В данных, хранящихся в s1 и s2, не встречается код "00h". (код ошибки 4101).

**ПРИМЕЧАНИЯ**

Команда сравнения строковых величин одновременно проверяет область данных, предусмотренную для сохранения. Однако даже если записанная строковая величина выходит за предел этой области, но различие обнаружено в предусмотренной области хранения строковой величины, результат команды выдается без сообщения об ошибке.

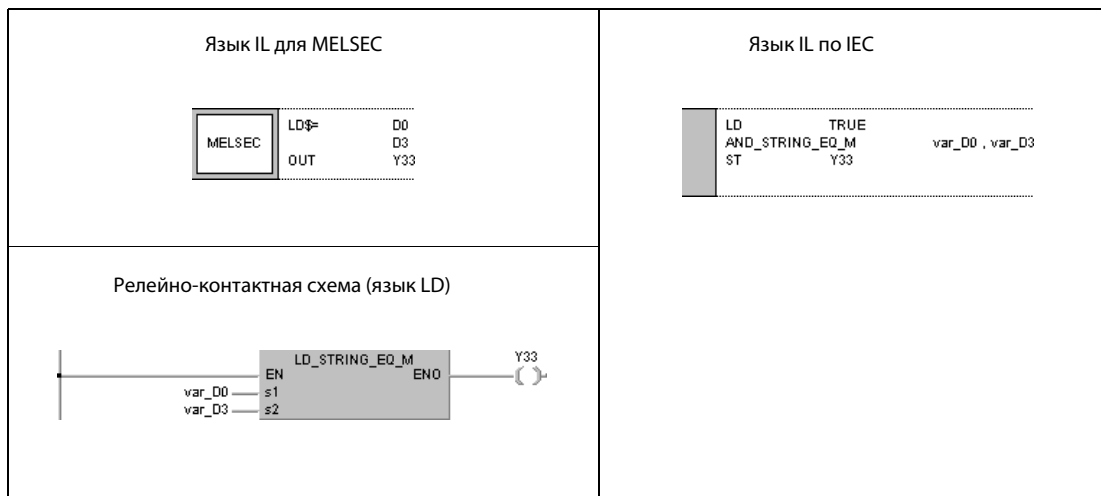


В вышеприведенном примере старшие 16 битов (D12288) хранящейся в s1 строковой величины находятся вне области данных, предусмотренной для сохранения, и программа переименовывает их в W0. Однако в связи с тем, что второй знак хранящейся в s1 строковой величины отличается от знака величины в s2, операция сравнения выдает результат 0 и сообщение об ошибке не выводится.

Без объявления переменных в заголовке программного компонента (POU) в GX IEC Developer эти программы не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительная информация имеется в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

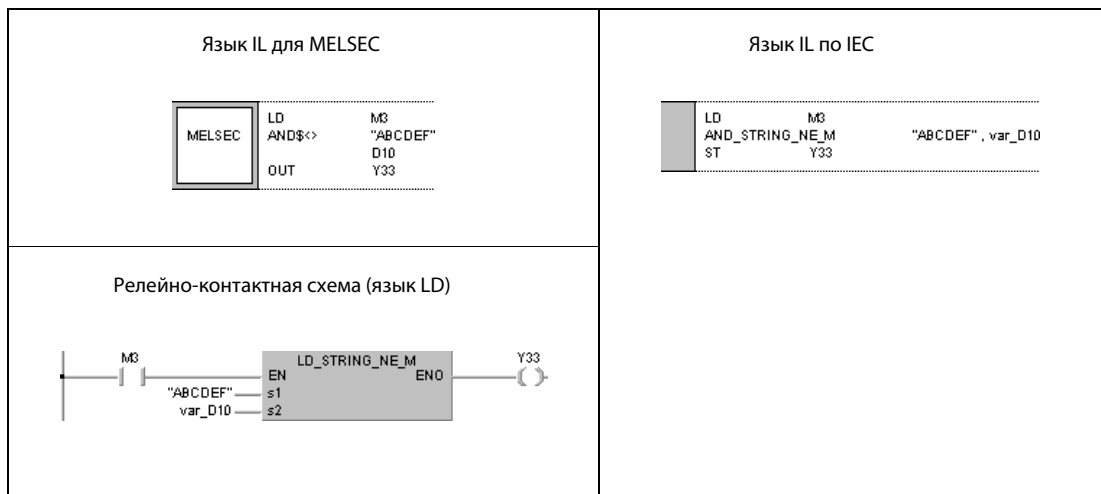
**Пример 1** Команда сравнения \$ =

Следующая программа сравнивает строковые величины в D0 со строковыми величинами в D3. Если эти значения равны, устанавливается Y33.



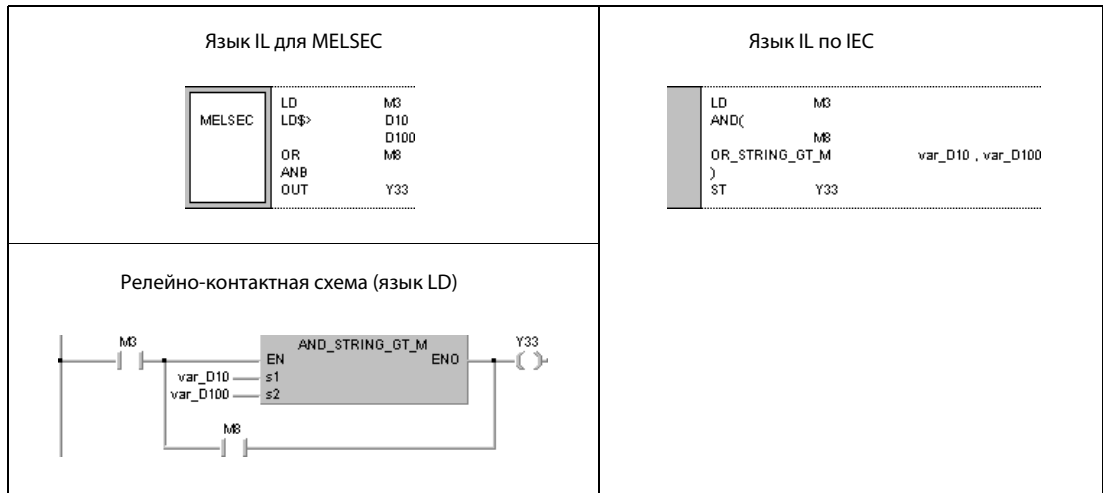
**Пример 2** Команда сравнения \$ <>

Следующая программа сравнивает строковую величину "ABCDEF" со строковой величиной, хранящейся в D10. Если содержимое в D10 отличается от "ABCDEF", устанавливается Y33.



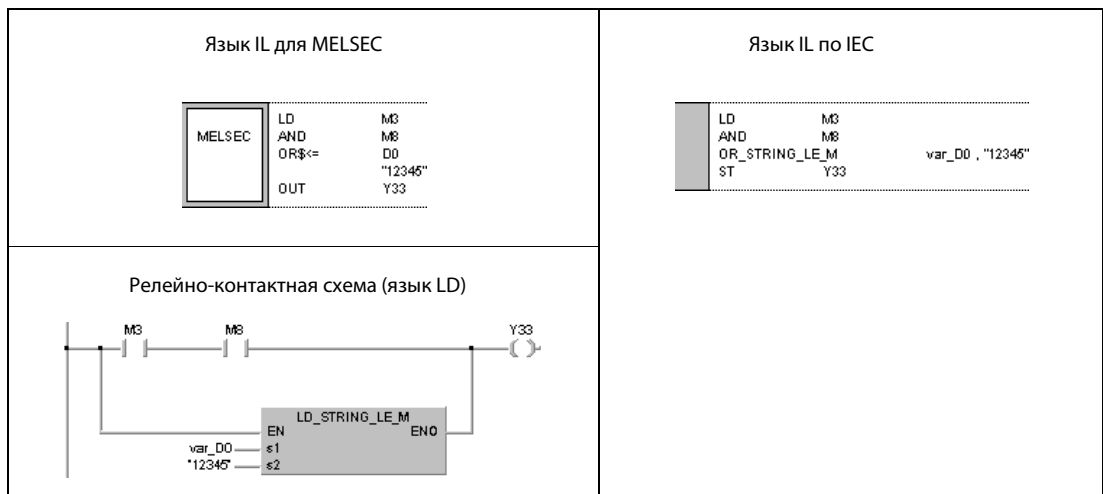
**Пример 3** Команда сравнения \$ >

Следующая программа сравнивает строковую величину в D10 со строковой величиной в D100. Если строковая величина в D10 больше, устанавливается Y33.



**Пример 4** Команда сравнения \$ < =

Следующая программа сравнивает строковую величину в D0 со строковой величиной "12345". Если строковая величина в D0 меньше, устанавливается Y33.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке программного компонента (POU) в GX IEC Developer эти программы не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительная информация имеется в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

6.1.5 БКСМР, БКСМРР

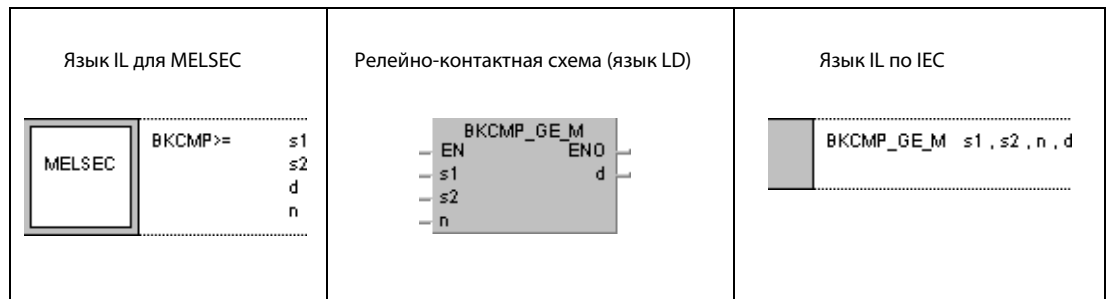
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

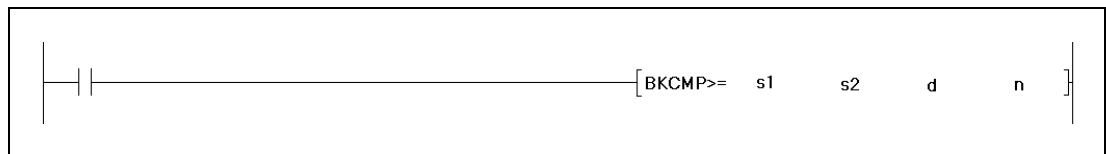
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	●	—	SM0	5
s2	—	●	●	—	—	—	—	—	—		
d	●	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
s1	Сравниваемые данные либо первый адрес, начиная с которого сохранены сравниваемые данные.	BIN, 16 бит
s2	Первый адрес, начиная с которого сохранены сравниваемые данные.	BIN, 16 бит
d	Первый адрес, начиная с которого должны сохраняться данные результата сравнения.	бит
n	Количество сравниваемых адресов (длина блока данных).	BIN, 16 бит

**Принцип действия**

**Поблочное сравнение двоичных данных**

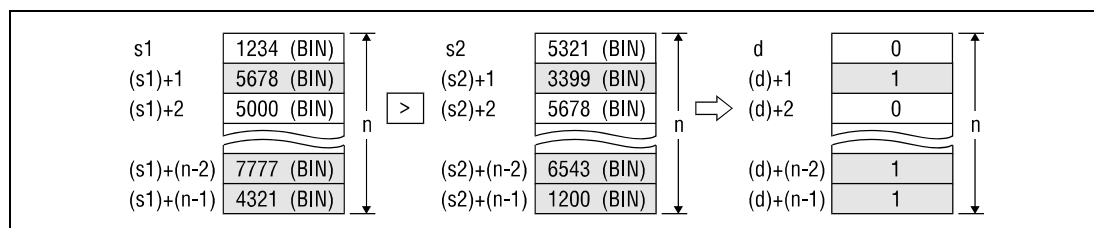
**ВКСМР Операторы сравнения**

Команда сравнения блоков двоичных данных состоит из самой команды, данных  $s1$  и  $s2$ , которые требуется сравнить друг с другом, обозначения цели  $d$ , в которой сохраняются результаты, и количества  $n$  сравниваемых блоков данных.

Сравнивается  $n$ -ый 16-битный блок адреса  $s1$  с  $n$ -ым 16-битным блоком адреса  $s2$ , начиная с младшего 16-битного блока. Результат каждого сравнения блоков сохраняется в  $d$ .

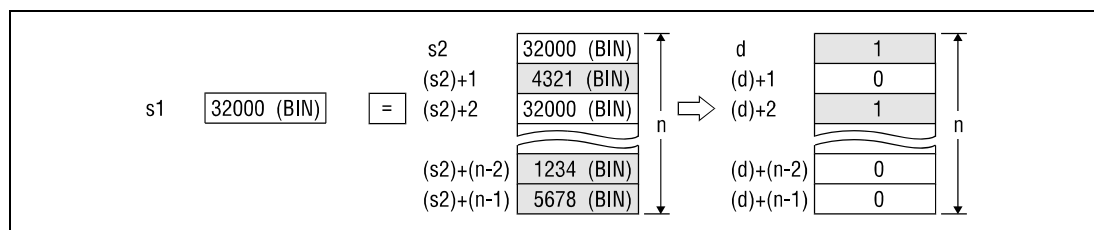
Если результат сравнения блока равен 1, в  $d$  записывается 1.

Если результат сравнения блока равен 0, в  $d$  записывается 0.



Операции сравнения выполняются единицами по 16 битов.

Хранящаяся в  $s1$  константа должна представлять собой 16-битное двоичное число в диапазоне между  $-32768$  и  $32767$ .



Результаты сравнений, выполняемых командами, перечислены в нижеследующей таблице.

Символ команды	Состояние выхода для $n$ -ного 16-битного блока	
	1, если:	0, если:
ВКСМР =	$s1 = s2$	$s1 \neq s2$
ВКСМР <>	$s1 \neq s2$	$s1 = s2$
ВКСМР >	$s1 > s2$	$s1 \leq s2$
ВКСМР <=	$s1 \leq s2$	$s1 > s2$
ВКСМР <	$s1 < s2$	$s1 \geq s2$
ВКСМР >=	$s1 \geq s2$	$s1 < s2$

Если все сохраненные в  $d$  результаты сравнения блоков имеют значение 1, устанавливается сигнал сравнения блоков SM704.

Если перед записью результата сравнения биты в  $d$  уже имели значение 1, они сохраняют это значение. Поэтому в случае применения команды ВКСМР\_Р перед повторной записью результатов сравнения (например, после изменения данных  $s1$  и  $s2$ ) биты в  $d$  следует заполнять нулями.

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в n количество блоков данных в s1, s2 и d находится вне предусмотренной для сохранения области операндов (MELSEC Q: код ошибки 4101).
- Область [от s1 до (s1) + (n - 1)] перекрывается с областью [от d до (d) + (n - 1)] (серии "Q"/"System Q": код ошибки 4101).
- Область [от s2 до (s2) + (n - 1)] перекрывается с областью [от d до (d) + (n - 1)] (серии "Q"/"System Q": код ошибки 4101).
- Область [от s1 до (s1) + (n - 1)] перекрывается с областью [от s2 до (s2) + (n - 1)] (серии "Q"/"System Q": код ошибки 4101).

**Пример 1**

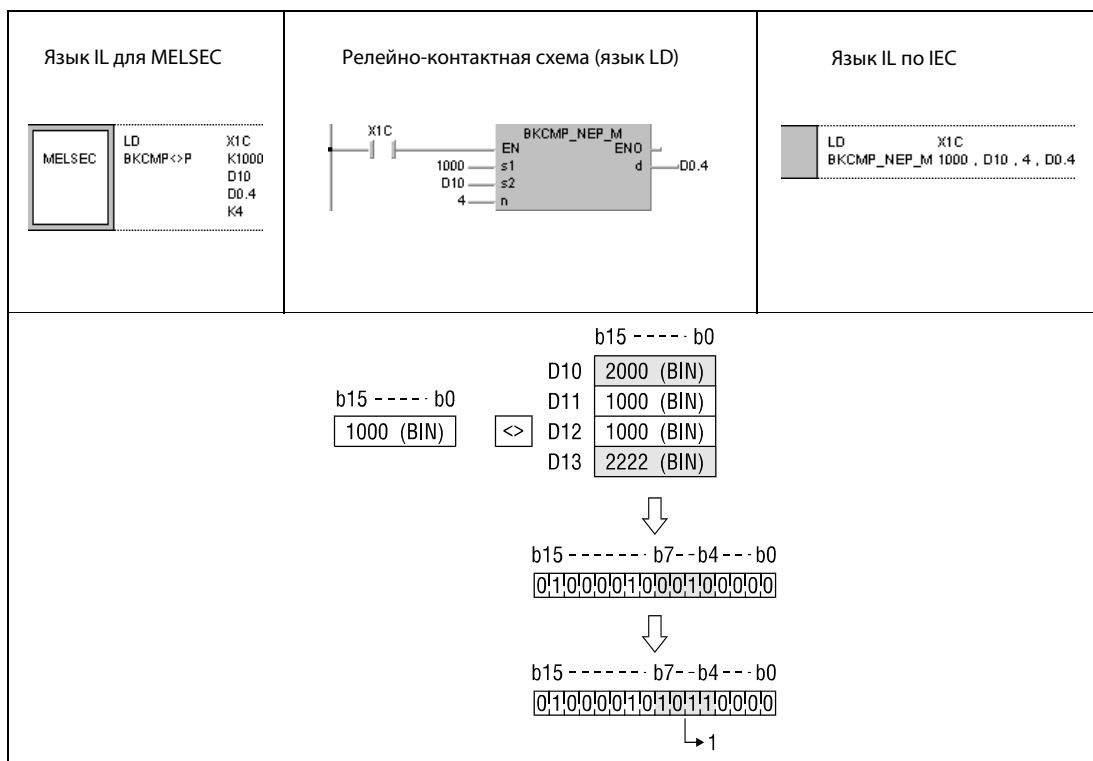
Команда сравнения ВКСМР = Р

Следующая программа при положительном (нарастающем) фронте сигнала X20 сравнивает блоки данных, начиная с D100, с блоками данных, начиная с R0. Результаты сравнения сохраняются, начиная с M10. Количество сравниваемых блоков (4) заложено в D0.

<p style="text-align: center;">Язык IL для MELSEC</p> <pre style="border: 1px solid black; padding: 5px;"> MELSEC LD      X20 BKCMR=P D100         R0         M10         D0                 </pre>	<p style="text-align: center;">Релейно-контактная схема (язык LD)</p>	<p style="text-align: center;">Язык IL по IEC</p> <pre style="border: 1px solid black; padding: 5px;"> LD      X20 BKCMR_EQP_M D100, R0, D0, M10                 </pre>																														
<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">b15 ----- b0</td> <td></td> <td style="text-align: center;">b15 ----- b0</td> <td></td> <td style="text-align: center;">b15 ----- b0</td> </tr> <tr> <td style="text-align: center;">D100</td> <td style="border: 1px solid black; padding: 2px;">1000 (BIN)</td> <td style="text-align: center;">R0</td> <td style="border: 1px solid black; padding: 2px;">1000 (BIN)</td> <td style="text-align: center;">M10</td> </tr> <tr> <td style="text-align: center;">D101</td> <td style="border: 1px solid black; padding: 2px;">2000 (BIN)</td> <td style="text-align: center;">R1</td> <td style="border: 1px solid black; padding: 2px;">2000 (BIN)</td> <td style="text-align: center;">M11</td> </tr> <tr> <td style="text-align: center;">D102</td> <td style="border: 1px solid black; padding: 2px;">3000 (BIN)</td> <td style="text-align: center;">R2</td> <td style="border: 1px solid black; padding: 2px;">5000 (BIN)</td> <td style="text-align: center;">M12</td> </tr> <tr> <td style="text-align: center;">D103</td> <td style="border: 1px solid black; padding: 2px;">4000 (BIN)</td> <td style="text-align: center;">R3</td> <td style="border: 1px solid black; padding: 2px;">4000 (BIN)</td> <td style="text-align: center;">M13</td> </tr> <tr> <td style="text-align: center;">D0</td> <td style="border: 1px solid black; padding: 2px;">4</td> <td></td> <td></td> <td></td> </tr> </table> <p style="text-align: center; margin-top: 10px;">=      ⇨</p>			b15 ----- b0		b15 ----- b0		b15 ----- b0	D100	1000 (BIN)	R0	1000 (BIN)	M10	D101	2000 (BIN)	R1	2000 (BIN)	M11	D102	3000 (BIN)	R2	5000 (BIN)	M12	D103	4000 (BIN)	R3	4000 (BIN)	M13	D0	4			
b15 ----- b0		b15 ----- b0		b15 ----- b0																												
D100	1000 (BIN)	R0	1000 (BIN)	M10																												
D101	2000 (BIN)	R1	2000 (BIN)	M11																												
D102	3000 (BIN)	R2	5000 (BIN)	M12																												
D103	4000 (BIN)	R3	4000 (BIN)	M13																												
D0	4																															

**Пример 2** Команда сравнения ВКСМР <> Р

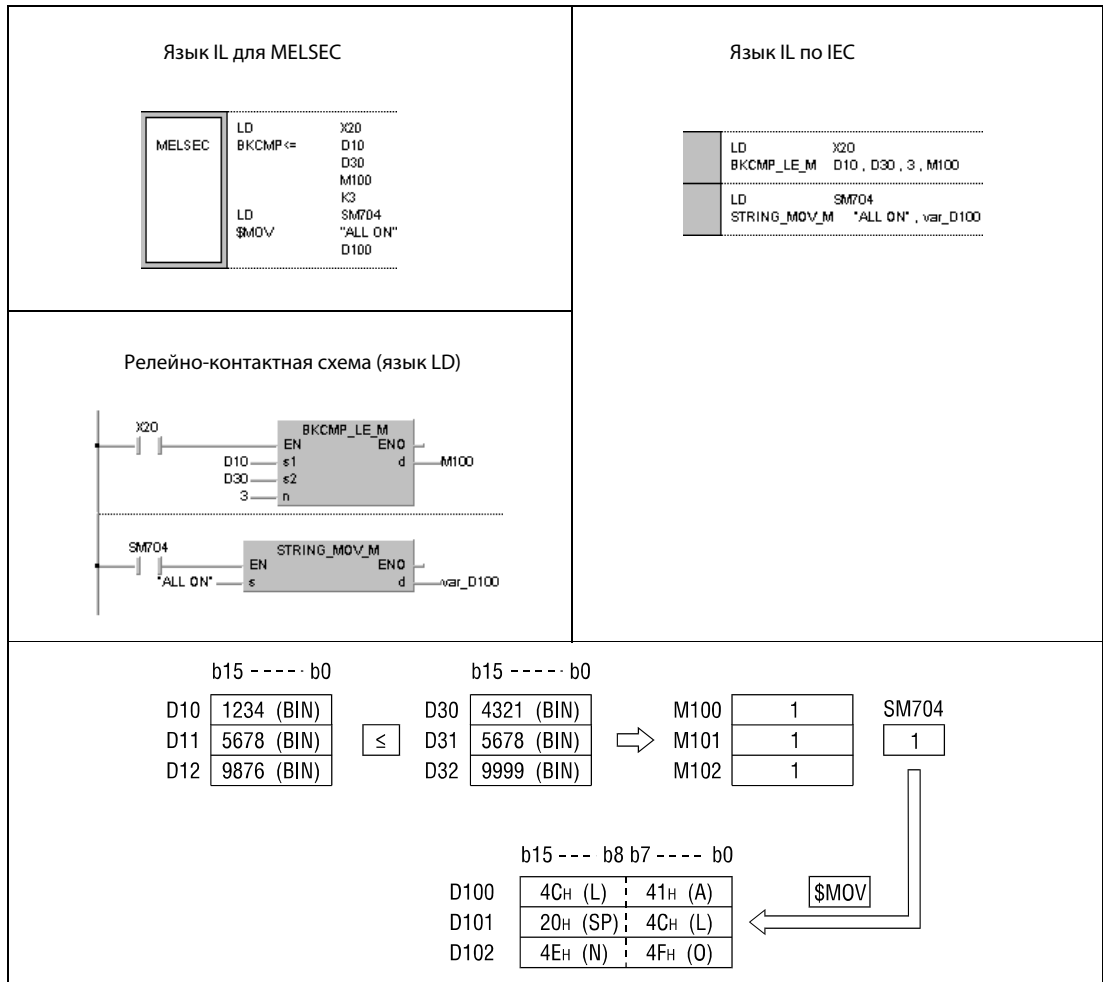
Следующая программа при положительном фронте сигнала X1C сравнивает константы K1000 с блоками данных, начиная с D10. Количество сравниваемых блоков (4) указывает константа K4. Результаты сравнения сохраняются в D0 от бита b4 до бита b7.



<sup>1</sup> Состояние этих двух битов после операции не изменяется (см. "Принцип действия").

**Пример 3** Команда сравнения ВКСМР <=

Следующая программа выполняет на время включенного состояния X20 сравняет блоки данных, начиная с D10, с блоками данных, начиная с D30. Количество сравниваемых блоков (3) указывает константа К3. Результаты сравнения сохраняются, начиная с M100. Если все результаты в M100 имеют значение 1, то устанавливается сигнал сравнения блоков SM704 и в адрес D100 передается строковая величина "ALL ON".



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке программного компонента (POU) в GX IEC Developer эта программа не работоспособна. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительная информация имеется в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.



## 6.2 Арифметические операции

Арифметические операции – это команды для выполнения простых математических действий (сложение, вычитание, умножение и деление) над двумя блоками данных.

В общей сложности имеются 54 (серии "Q" и "System Q") или 40 (серия "A") различных арифметических функций.

Функция	BIN		BCD		
	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC	
+	Сложение	+	PLUS_M, PLUS_3_M	B+	BPLUS_M, BPLUS_3_M
		+P	PLUSP_M, PLUSP_3_M	B+P	BPLUSP_M, BPLUSP_3_M
		D+	DPLUS_M, DPLUS_3_M	DB+	DBPLUS_M, DBPLUS_3_M
		D+P	DPLUSP_M, DPLUSP_3_M	DB+P	DBPLUSP_M, DBPLUSP_3_M
-	Вычитание	-	MINUS_M, MINUS_3_M	B-	BMINUS_M, BMINUS_3_M
		-P	MINUSP_M, MINUSP_3_M	B-P	BMINUSP_M, BMINUSP_3_M
		D-	DMINUS_M, DMINUS_3_M	DB-	DBMINUS_M, DBMINUS_3_M
		D-P	DMINUSP_M, DMINUSP_3_M	DB-P	DBMINUSP_M, DBMINUSP_3_M
×	Умножение	×	MULTI_3_M	B×	BMULTI_M
		×P	MULTIP_3_M	B×P	BMULTIP_M
		D×	DMULTI_3_M	DB×	DBMULTI_M
		D×P	DMULTIP_3_M	DB×P	DBMULTIP_M
/	Деление	/	DIVID_3_M	B/	BDIVID_M
		/P	DIVIDP_3_M	B/P	BDIVIDP_M
		D/	DDIVID_3_M	DB/	DBDIVID_M
		D/P	DDIVIDP_3_M	DB/P	DBDIVIDP_M
+1	Приращение	INC	INC_M		
		INCP	INCP_M		
		DINC	DINC_M		
		DINCP	DINCP_M		
-1	Декремент	DEC	DEC_M		
		DECP	DECP_M		
		DDEC	DDEC_M		
		DDECP	DDECP_M		

**ПРИМЕЧАНИЕ** В редакторах IEC следует использовать команды по стандарту IEC.

Функция	Числа с плавающей запятой		Блок двоичных данных	
	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC
+	E +	EPLUS_M, EPLUS_3_M	BK +	BKPLUS_M
	E + P	EPLUSP_M, EPLUSP_3_M	BK + P	BKPLUSP_M
-	E -	EMINUS_M, EMINUS_3_M	BK -	BKMINUS_M
	E - P	EMINUSP_M, EMINUSP_3_M	BK - P	BKMINUSP_M
×	E ×	EMUL_M		
	E × P	EMULP_M		
/	E /	EDIV_M		
	E / P	EDIVP_M		

Функция	Строковые величины	
	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC
+	\$ +	STRING_PLUS_M, STRING_PLUS_3_M
	\$ + P	STRING_PLUSP_M, STRING_PLUSP_3_M

Арифметические операции для чисел с плавающей запятой, двоичных блоков данных и строковых величин действительны только для серии "Q".

### Арифметические операции над двоичными данными

Если результат сложения превышает 32767 (2147483647 в случае 32-битной команды), результат становится отрицательным.

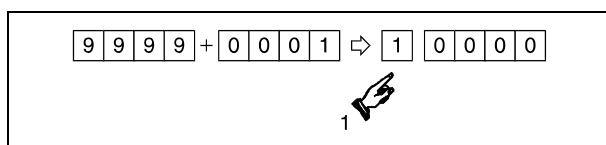
Если результат вычитания снижается ниже -32768 (-2147483648 в случае 32-битной команды), результат становится положительным.

Положительные и отрицательные значения вычисляются следующим образом:

$$\begin{aligned}
 5 + 8 &= 13 \\
 5 - 8 &= -3 \\
 5 \times 3 &= 15 \\
 -5 \times 3 &= -15 \\
 -5 \times (-3) &= 15 \\
 5 / 3 &= 1 \text{ с остатком } 2 \\
 -5 / 3 &= -1 \text{ с остатком } -2 \\
 5 / (-3) &= -1 \text{ с остатком } 2 \\
 -5 / (-3) &= 1 \text{ с остатком } -2
 \end{aligned}$$

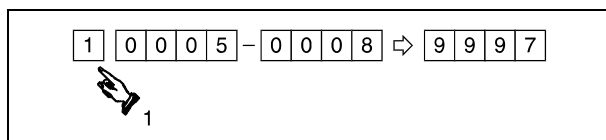
### Арифметические операции над двоично-десятичными данными

Если результат сложения превышает 9999 (99999999 в случае 32-битной команды), старшие биты игнорируются (переполнение). Флаг переноса в этом случае не устанавливается.



<sup>1</sup> Перенос игнорируется.

Если уменьшаемое меньше вычитаемого, перенос выполняется следующим образом:



<sup>1</sup> перенос

6.2.1 + , + P , - , - P

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

	Операнды																			Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011		
	битовые							словные (16 бит)							Константы		Указатели		Уровень							
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P						I	N
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	5			
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●								1			
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					7	●		●
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					1			
d1		●	●	●	●	●	●	●	●	●	●	●	●	●	●											

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

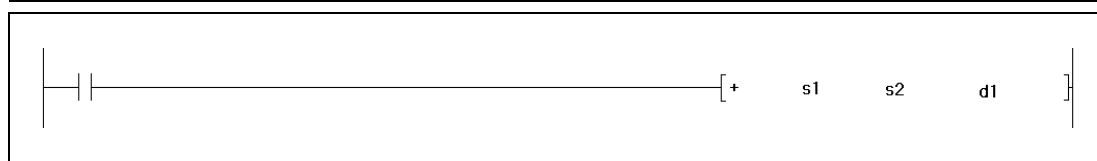
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Специальные модули U□\G□□	Индексные регистры Zп	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	●	—	—	3
d	●	●	●	●	●	●	●	—	—		
s1	●	●	●	●	●	●	●	●	—	—	4
s2	●	●	●	●	●	●	●	●	—	—	
d1	●	●	●	●	●	●	●	—	—	—	

GX IEC Developer

<p>Язык IL для MELSEC</p> <pre> MELSEC \$+      s MELSEC \$+      s1                  s2                  d1                     </pre>	<p>Релейно-контактная схема (язык LD)</p>	<p>Язык IL по IEC</p> <pre> PLUS_3_M      s1 , s2 , d1                     </pre>
---	---	---

GX Developer



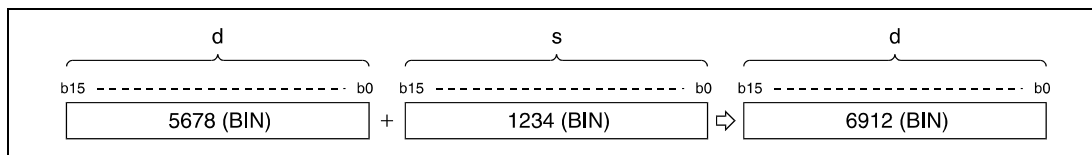
Переменные

Операнд	Значение	Тип данных
s	Слагаемые или вычитаемые данные либо первый адрес, начиная с которого сохранены эти данные.	BIN, 16 бит
d	Слагаемые или уменьшаемые данные либо первый адрес, начиная с которого сохранены эти данные, а также первый адрес, начиная с которого сохраняется сумма или разность.	
s1	Слагаемые или уменьшаемые данные либо первый адрес, начиная с которого сохранены эти данные.	
s2	Слагаемые или вычитаемые данные либо первый адрес, начиная с которого сохранены эти данные.	
d1	Первый адрес, начиная с которого сохраняется сумма или разность.	

**Принцип действия****Сложение и вычитание двоичных данных (16 бит)****+ Сложение двоичных данных (16 бит)**

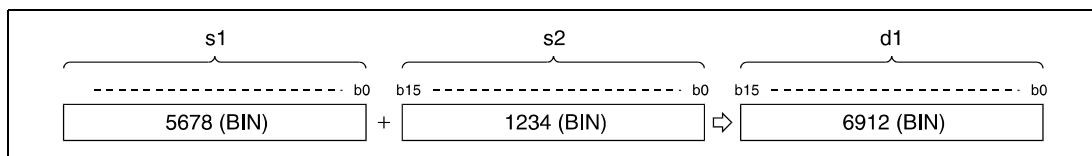
## ● 1-й вариант:

Указанные в d двоичные данные суммируются с двоичными данными в s. Результат сложения сохраняется в d.



## ● 2-й вариант:

Указанные в s1 двоичные данные суммируются с двоичными данными в s2. Результат сложения выводится в операнд, указанный в d1.



Указанное в s, d, s1, s2 и d1 значение должно представлять собой 16-битное двоичное число в диапазоне между -32768 и 32767.

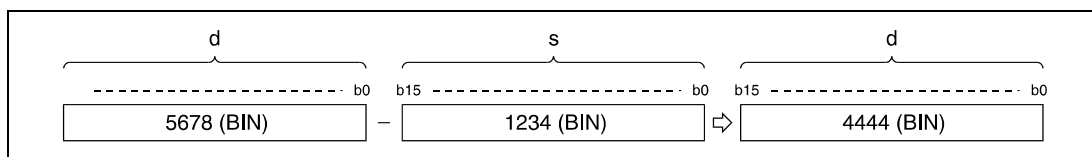
Самый старший бит (b15) устанавливает арифметический знак значений данных в s, d, s1, s2 и d1. Если они положительные, этот бит = 0. Если они отрицательные, этот бит = 1.

При занижении самого младшего бита (b0) или превышении самого старшего бита (b15) флаг переноса не устанавливается.

**- Вычитание двоичных данных (16 бит)**

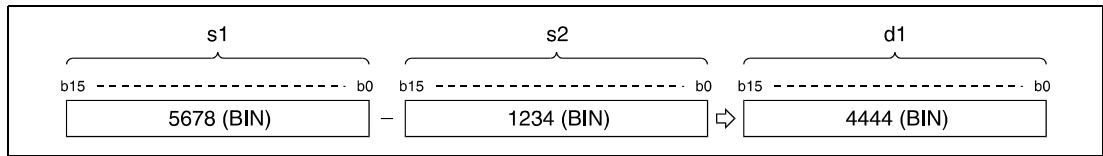
## ● 1-й вариант:

Указанные в s двоичные данные вычитаются из двоичных данных в d. Разность сохраняется в d.



● 2-й вариант:

Указанные в s2 двоичные данные вычитаются из двоичных данных в s1. Результат выводится в операнд, указанный в d1.



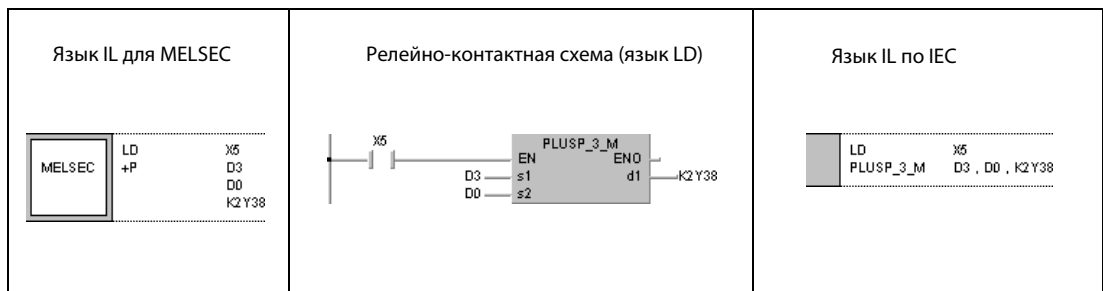
Указанное в s, d, s1, s2 и d1 значение должно представлять собой 16-битное двоичное число в диапазоне между -32768 и 32767.

Самый старший бит (b15) устанавливает арифметический знак значений данных в s, d, s1, s2 и d1. Если они положительные, этот бит = 0. Если они отрицательные, этот бит = 1.

При занижении самого младшего бита (b0) или превышении самого старшего бита (b15) флаг переноса не устанавливается.

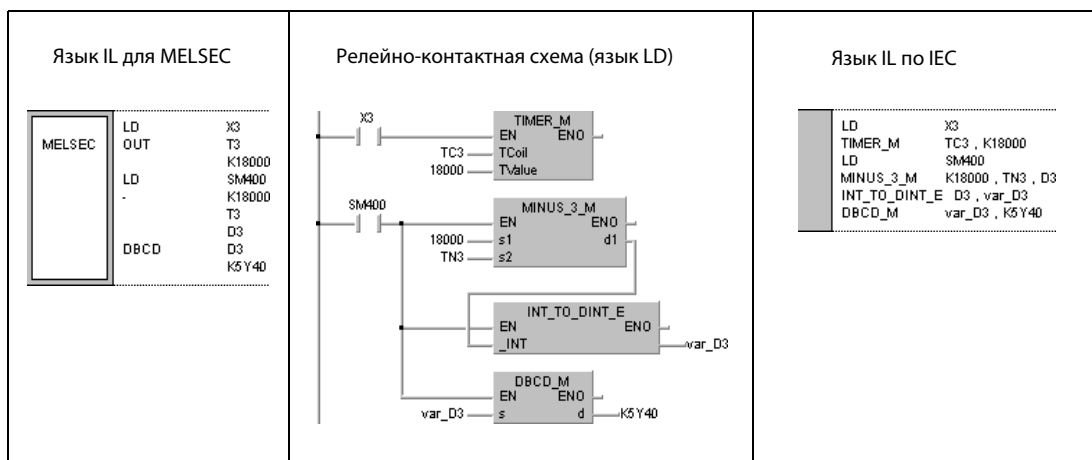
**Пример 1** + P

Следующая программа при положительном фронте сигнала X5 прибавляет содержимое D3 к содержимому D0 и выдает результат по адресу Y38...Y3F.



**Пример 2** –

Следующая программа выводит разность между заданным и фактическим значением таймера T3 по адресу Y40...Y53 в формате BCD.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительная информация имеется в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

6.2.2 D +, D + P, D -, D - P

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

	Операнды															Указа-тели	Уро-вень	Длина блока	Шагов	Индекс	Флаг пере-носа	Флаг оши-бки					
	битовые							словные (16 бит)							Константы												
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z								V	K	H (16#)	P	I
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K8	9				
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●								11				●
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					11				●
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					11				●
d1		●	●	●	●	●	●	●	●	●	●	●	●	●	●							11				●	

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

Операнды MELSEC Q

	Операнды										Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□\G□	Индексные регистры Zп	Константы K, H (16#)	Иные			
	битовые	словные		битовые	словные							
s	●	●	●	●	●	●	●	●	—	—	—	3 <sup>1)</sup>
d	●	●	●	●	●	●	●	—	—	—		
s1	●	●	●	●	●	●	●	●	—	—	4 <sup>2)</sup>	
s2	●	●	●	●	●	●	●	●	—	—		
d1	●	●	●	●	●	●	●	—	—	—		

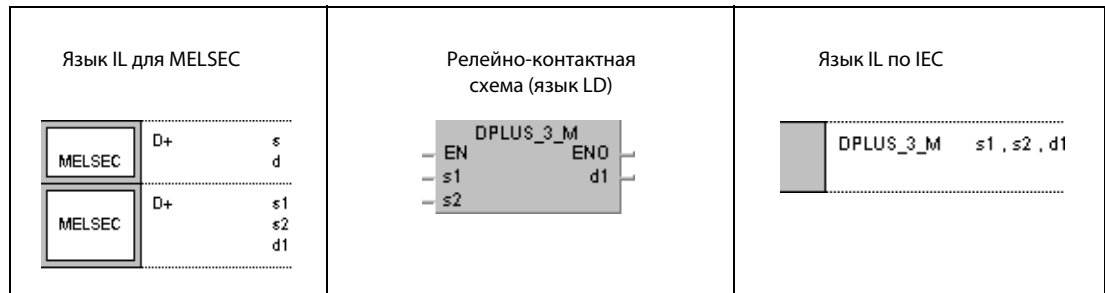
<sup>1</sup> Количество шагов программы зависит от типа центрального процессора и используемых операндов.

- При использовании процессоров QnA или однопроцессорного модуля ЦП серии "Q": 3
- При использовании многопроцессорного модуля ЦП серии "Q" и внутренних словных операндов (кроме регистра файлов ZR): 5
- Константы: 5
- Битовые операнды с адресом, кратным 16, имеющие обозначение битового блока K8 и не индексируемые: 5
- При использовании многопроцессорного модуля ЦП серии "Q" и иных операндов кроме вышеуказанных: 3

<sup>2</sup> Количество шагов программы зависит от типа центрального процессора и используемых операндов.

- При использовании процессоров QnA или однопроцессорного модуля ЦП серии "Q": 4
- При использовании многопроцессорного модуля ЦП серии "Q" и внутренних словных операндов (кроме регистра файлов ZR): 6
- Константы: 6
- Битовые операнды с адресом, кратным 16, имеющие обозначение битового блока K8 и не индексируемые: 6
- При использовании многопроцессорного ЦПУ серии "Q" и иных операндов кроме вышеуказанных: 4

GX IEC Developer





**GX Developer**



**Переменные**

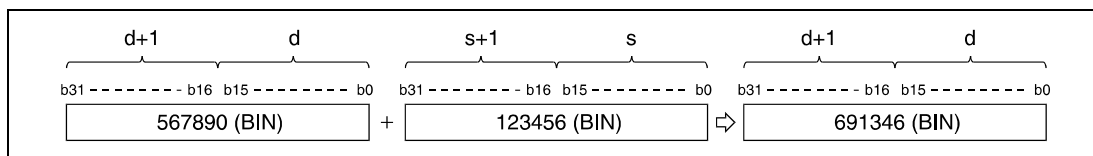
BIN, 64 бита

Операнд	Значение	Тип данных
s	Слагаемые или вычитаемые данные либо первый адрес, начиная с которого сохранены эти данные.	BIN, 32 бита
d	Слагаемые или уменьшаемые данные либо первый адрес, начиная с которого сохранены эти данные, а также первый адрес, начиная с которого сохраняется сумма или разность.	
s1	Слагаемые или уменьшаемые данные либо первый адрес, начиная с которого сохранены эти данные.	
s2	Слагаемые или вычитаемые данные либо первый адрес, начиная с которого сохранены эти данные.	
d1	Первый адрес, начиная с которого сохраняется сумма или разность.	

**Принцип действия****Сложение и вычитание двоичных данных (32 бита)****D + Сложение двоичных данных (32 бита)**

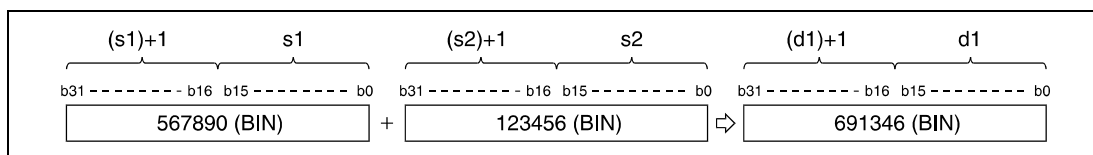
## ● 1-й вариант:

Указанные в d двоичные данные суммируются с двоичными данными в s. Результат сложения сохраняется в d.



## ● 2-й вариант:

Указанные в s1 двоичные данные суммируются с двоичными данными в s2. Результат сложения выводится в операнд, указанный в d1.



Указанное в s, d, s1, s2 и d1 значение данных должно представлять собой 32-битное двоичное число в диапазоне между -2147483648 и 2147483647.

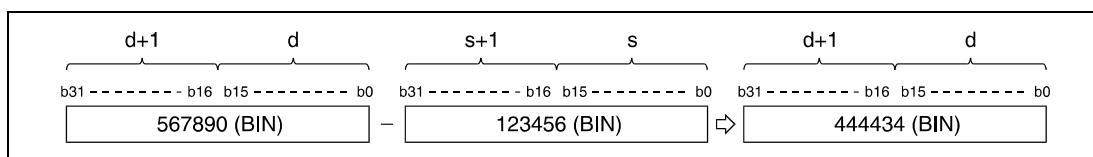
Последний бит (b31) устанавливает арифметический знак значений данных в s, d, s1, s2 и d1. Если они положительные, этот бит = 0. Если они отрицательные, этот бит = 1.

При занижении самого младшего бита (b0) или превышении самого старшего бита (b31) флаг переноса не устанавливается.

**D - Вычитание двоичных данных (32 бита)**

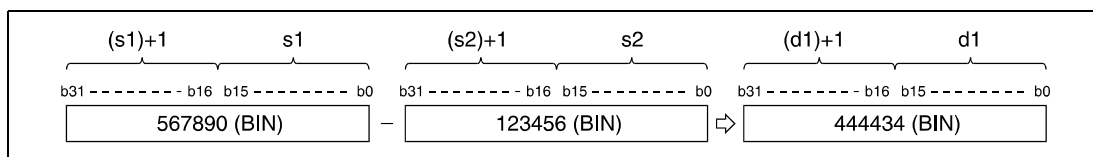
## ● 1-й вариант:

Указанные в s двоичные данные вычитаются из двоичных данных в d. Разность сохраняется в d.



● 2-й вариант:

Указанные в s2 двоичные данные вычитаются из двоичных данных в s1. Разность выводится в операнд, указанный в d1.



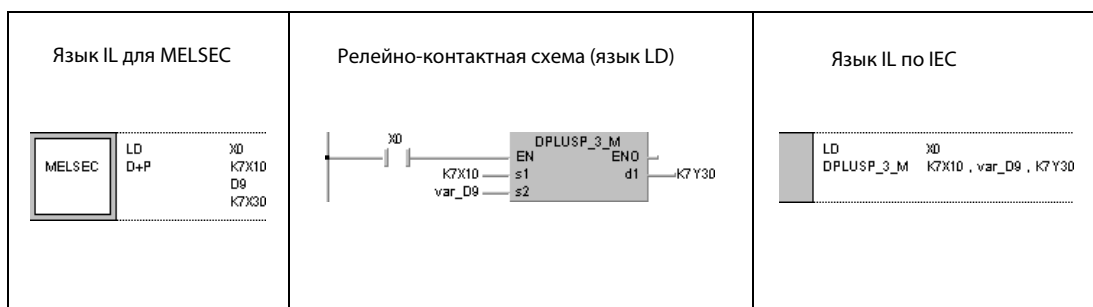
Указанное в s, d, s1, s2 и d1 значение данных должно представлять собой 32-битное двоичное число в диапазоне между -2147483648 и 2147483647.

Последний бит (b31) устанавливает арифметический знак значений данных в s, d, s1, s2 и d1. Если они положительные, этот бит = 0. Если они отрицательные, этот бит = 1.

При занижении самого младшего бита (b0) или превышении самого старшего бита (b31) флаг переноса не устанавливается.

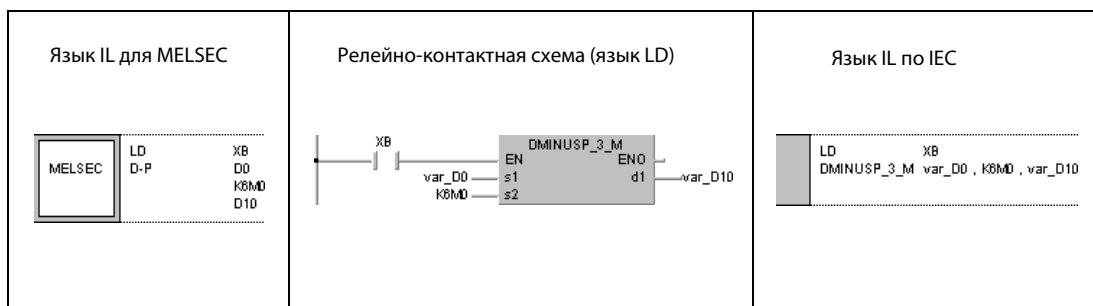
**Пример 1** D + P

Следующая программа при положительном фронте сигнала X0 прибавляет содержимое операндов X10...X2B к содержимому регистров D9 и D10 и выводит результат в операнды Y30...Y4B.



**Пример 2** D - P

Следующая программа при положительном фронте сигнала XB вычитает данные M0...M23 из D0 и D1 и выводит результат в D10 и D11.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительная информация имеется в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

6.2.3 x, xP, /, /P

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

	Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011
	битовые						словные (16 бит)						Константы		Указа-тели	Уро-вень					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V					
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				
d1		●	●	●	●	●	●	●	●	●	●	●									

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непр. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	●	●	●	●	●	●	●	●	—	SM0	4 <sup>1)</sup>
s2	●	●	●	●	●	●	●	●	—		
d1	●	●	●	●	●	●	●	—	—		

<sup>1</sup> Количество шагов программы зависит от типа центрального процессора и используемых операндов.

При использовании процессоров QnA: 4

При использовании процессора Q

и внутренних словных операндов (кроме регистра файлов ZR): 3

Константы: 3

Битовые операнды с адресом, кратным 16, имеющие обозначение битового блока K4

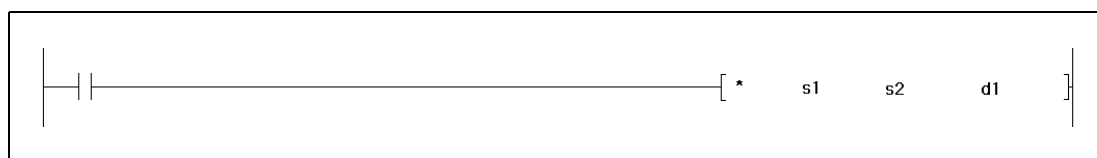
и не индексируемые: 3

При использовании процессора Q и иных операндов кроме вышеуказанных: 4

GX IEC Developer



GX Developer



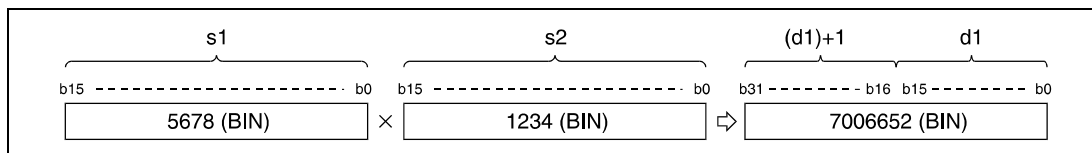
Переменные

Операнд	Значение	Тип данных
s1	Данные множителя или делимого либо первый адрес, начиная с которого сохранены эти данные.	BIN, 16 бит
s2	Данные множителя или делителя либо первый адрес, начиная с которого сохранены эти данные.	BIN, 16 бит
d1	Первый адрес, начиная с которого сохраняется произведение или частное.	BIN, 32 бита

BIN, 32 бита

**Принцип действия** **Умножение и деление двоичных данных (16 бит)****x** **Умножение двоичных данных (16 бит)**

Указанные в s1 двоичные данные умножаются на двоичные данные в s2 и произведение сохраняется в d1.



Если результат в d1 представляет собой битовый операнд, то адресация битов начинается с младших битов.

Пример:

K1: 4 младших бита (от b0 до b3)

K4: 16 младших битов (от b0 до b15)

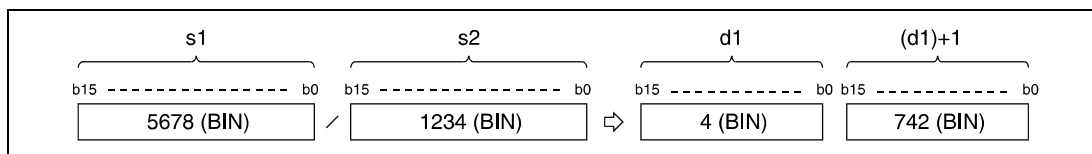
K8: 32 бита (от b0 до b31)

Указанное в s1 и s2 значение должно представлять собой 16-битное двоичное число в диапазоне между -32768 и 32767.

Самый старший бит (b15 или b31) в d1 устанавливает арифметический знак значений данных в s1, s2 или d1. Если они положительные, этот бит = 0. Если они отрицательные, этот бит = 1.

**/** **Деление двоичных данных (16 бит)**

Указанные в s1 двоичные данные делятся на двоичные данные в s2 и результат деления сохраняется в d1.



В случае словных операндов результат деления сохраняется в виде 32-битного значения, разделенного на частное и остаток деления. В битовых операндах можно сохранять только частное.

Частное сохраняется в младших 16 битах. Остаток сохраняется в старших 16 битах (только в случае словных операндов).

Указанное в s1 и s2 значение должно представлять собой 16-битное двоичное число в диапазоне между -32768 и 32767.

Самый старший бит (b15) устанавливает арифметический знак значений в s1, s2, d1 и (d1) + 1.

**Источники ошибок**

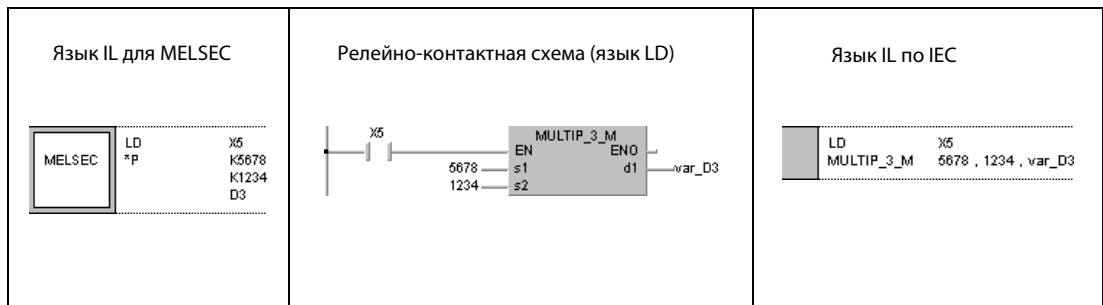
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В качестве d1 указан адрес A1 или V (серия "A").
- Делитель s2 равен 0 (серии "Q"/"System Q" = код ошибки 4100).

**Пример 1**

x P

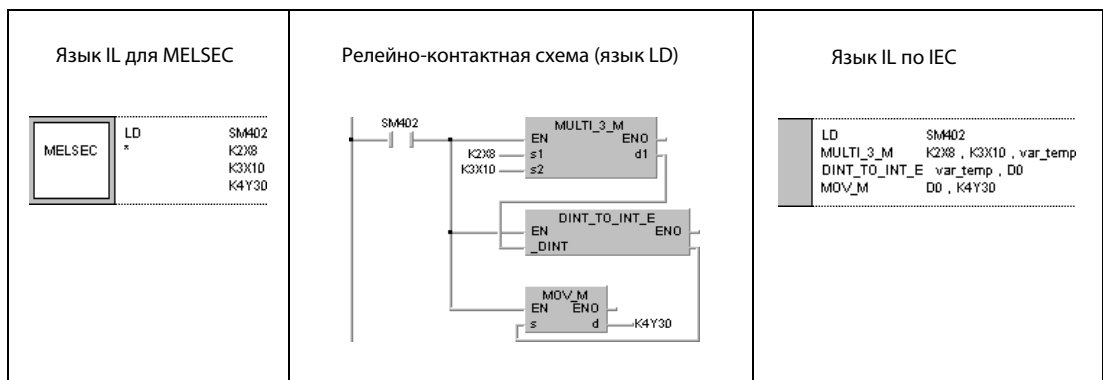
Следующая программа при положительном фронте сигнала X5 сохраняет произведение 5678 и 1234 в виде двоичного значения в D3 и D4.



**Пример 2**

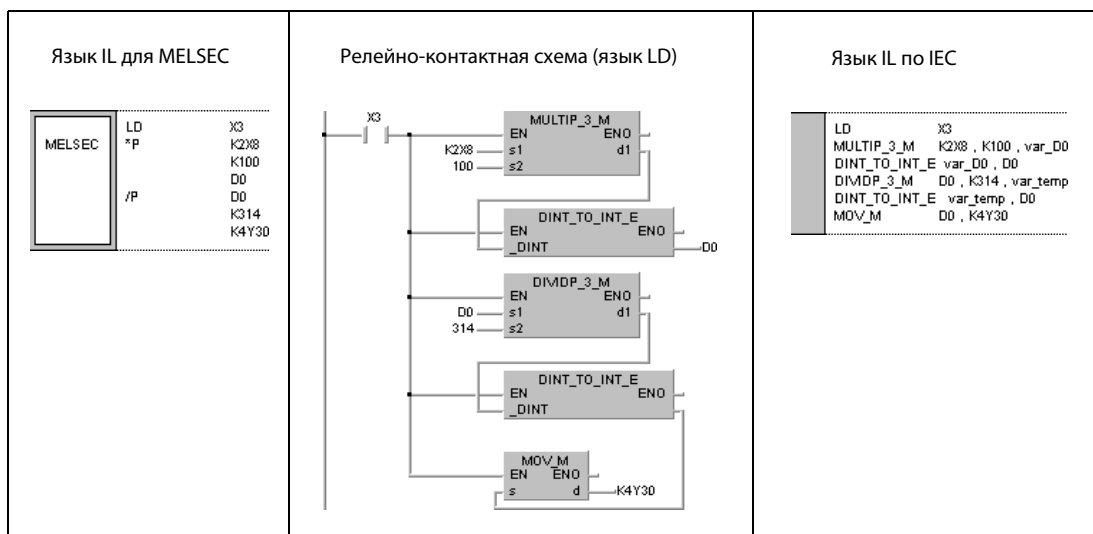
x

Следующая программа выводит произведение двоичных данных X8...XF и X10...X1B на выходы Y30...Y3F.



**Пример 3** /P

Следующая программа при положительном фронте сигнала X3 делит значение в X8...XF на 3.14. Результат выводится на Y30...Y3F.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительная информация имеется в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

6.2.4 D x, D x P, D/, D/P

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

	Операнды														Указа-тели	Уро-вень	Длина блока	Шагов	Индекс	Флаг пере-носа	Флаг ошибки				
	битовые							словные (16 бит)														Константы			
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1								Z	V	K	H (16#)
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K8	11 ↓ 1	●	●	●
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●									
d1	●	●	●	●	●	●	●	●	●	●	●	●	●	●											

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

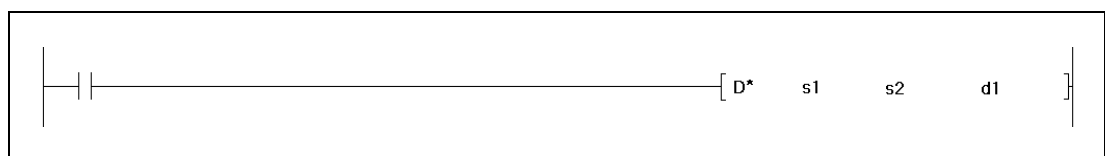
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 неоср. адр. J□□		Спец. модули U□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	●	●	●	●	●	●	●	●	—	SM0	4
s2	●	●	●	●	●	●	●	●	—		
d1	●	●	●	—	—	—	—	—	—		

GX IEC Developer



GX Developer



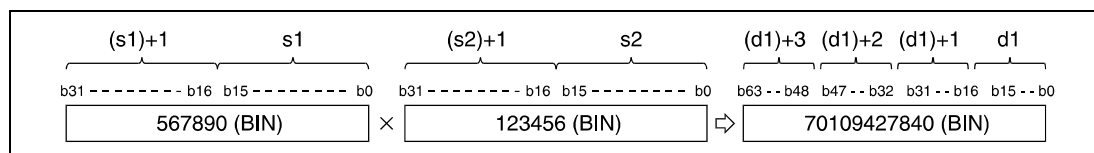
Переменные

Операнд	Значение	Тип данных	
		MELSEC	IEC
s1	Данные множителя или делимого либо первый адрес, начиная с которого сохранены эти данные.	BIN, 32 бита	ANY32
s2	Данные множителя или делителя либо первый адрес, начиная с которого сохранены эти данные.	BIN, 32 бита	ANY32
d1	Первый адрес, начиная с которого сохраняется произведение или частное.	BIN, 64 бита	массив [1..2] данных типа ANY32



**Принцип действия** **Умножение и деление двоичных данных (32 бита)****Dx** **Умножение двоичных данных (32 бита)**

Указанные в  $s1$  двоичные данные умножаются на двоичные данные в  $s2$  и произведение сохраняется в  $d1$ .



Если результат в  $d1$  представляет собой битовый операнд, то могут быть установлены только 32 младших бита.

Пример:

K1: 4 младших бита (от  $b0$  до  $b3$ )

K4: 16 младших битов (от  $b0$  до  $b15$ )

K8: 32 бита (от  $b0$  до  $b31$ )

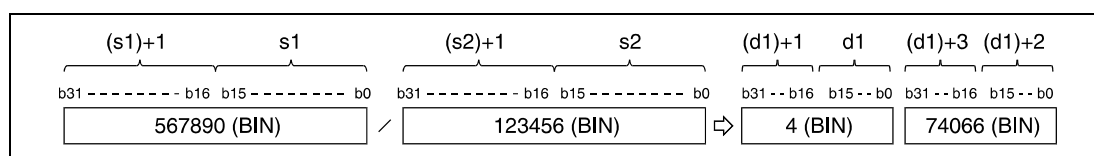
Если для записи результата умножения в битовый операнд нужны старшие 32 бита, то эти данные сначала должны быть сохранены в словном операнде. Затем данные  $(d1) + 2$  и  $(d1) + 3$  словного операнда передаются в указанный битовый операнд.

Указанное в  $s1$  и  $s2$  значение должен представлять собой 32-битное двоичное число в диапазоне между  $-2147483648$  и  $2147483647$ .

Самый старший бит ( $b31$  или  $b63$ ) в  $d1$  устанавливает арифметический знак данных в  $s1$ ,  $s2$  и  $d1$ . Если они положительные, этот бит = 0. Если они отрицательные, этот бит = 1.

**D/** **Деление двоичных данных (32 бита)**

Указанные в  $s1$  двоичные данные делятся на двоичные данные в  $s2$  и результат деления сохраняется в  $d1$ .



В случае словных операндов результат деления сохраняется в виде массива значений типа DINT (64 бита). При этом он разделяется на частное и остаток деления. В битовых операндах можно сохранять только частное.

Частное сохраняется в младших элементах массива (32 бита). Остаток сохраняется в старших элементах массива (32 бита) (только в случае словного операнда).

Указанное в  $s1$  и  $s2$  значение должен представлять собой 32-битное двоичное число в диапазоне между  $-2147483648$  и  $2147483647$ .

Старший бит ( $b31$ ) устанавливает арифметический знак значений в  $s1$ ,  $s2$ ,  $d1$  и  $(d1) + 2$ . Если они положительные, этот бит = 0. Если они отрицательные, этот бит = 1.

**Источники ошибок**

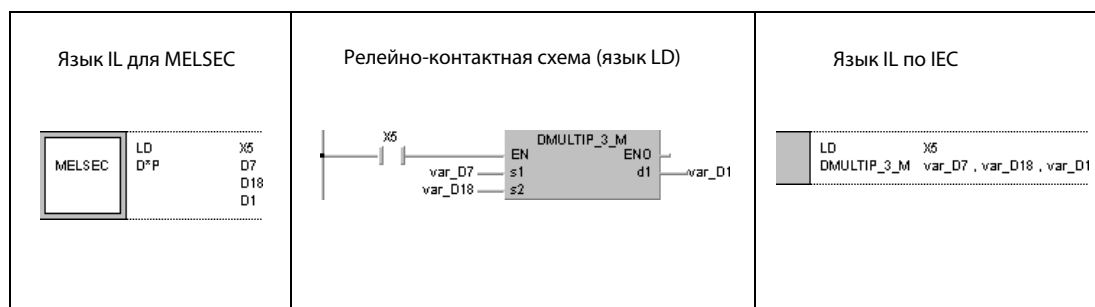
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В качестве s1 или s2 указан адрес A1 или V, а в качестве d1 – адрес A0, A1, Z или V (серия "A").
- Делитель s2 равен 0 (серии "Q"/"System Q" = код ошибки 4100).

**Пример 1**

D x P

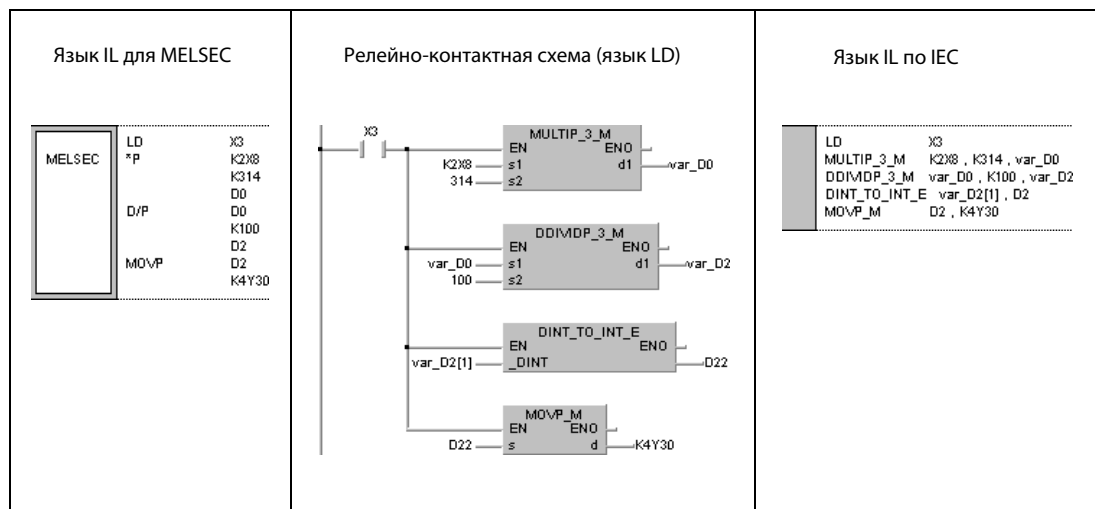
Следующая программа при положительном фронте сигнала X5 умножает двоичные данные в D7 и D8 на двоичные данные в D18 и D19 и сохраняет произведение в D1...D4.



**Пример 2**

x P

Следующая программа при положительном фронте сигнала X3 умножает данные в X8...XF на 3.14 и выводит результат на Y30...Y3F.



**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительная информация имеется в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

**6.2.5 B +, B + P, B -, B - P**

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

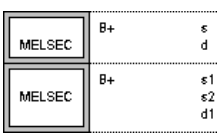
	Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011				
	битовые						словные (16 бит)						Константы		Указатели							Уровень			
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						K	H (16#)	P	I
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	7	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●	●						9					
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●									
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●									
d1		●	●	●	●	●	●	●	●	●	●	●	●	●											

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

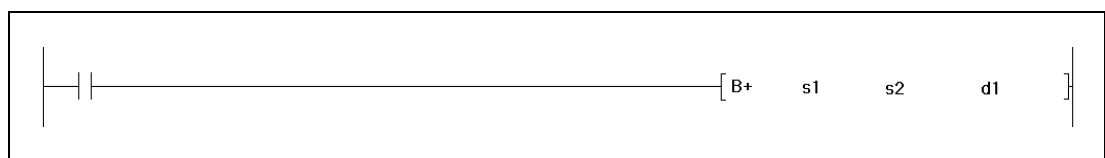
Операнды MELSEC Q

	Операнды							Флаг ошибки	Шагов		
	Внутренние операнды (системн., польз.ов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□\G□□	Индексные регистры Zn			Константы K, H (16#)	Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	●	—	SMO	3
d	●	●	●	●	●	●	—	—			
s1	●	●	●	●	●	●	●	●	—	SMO	4
s2	●	●	●	●	●	●	●	●	—		
d1	●	●	●	●	●	●	—	—			

GX IEC Developer

<p>Язык IL для MELSEC</p> 	<p>Релейно-контактная схема (язык LD)</p> 	<p>Язык IL по IEC</p> 
---	---	---

GX Developer



Переменные

Операнд	Значение	Тип данных
s	Слагаемые или вычитаемые данные либо первый адрес, начиная с которого сохранены эти данные.	BCD, 4 знака
d	Слагаемые или уменьшаемые данные либо первый адрес, начиная с которого сохранены эти данные, а также первый адрес, начиная с которого сохраняется сумма или разность.	
s1	Слагаемые или уменьшаемые данные либо первый адрес, начиная с которого сохранены эти данные.	
s2	Слагаемые или вычитаемые данные либо первый адрес, начиная с которого сохранены эти данные.	
d1	Первый адрес, начиная с которого сохраняется сумма или разность.	

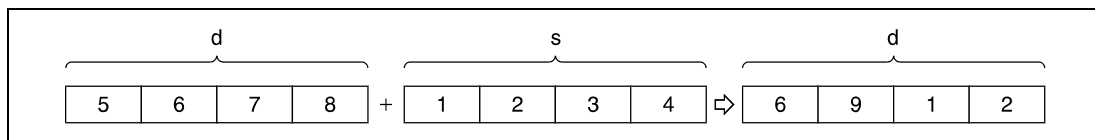
**Принцип действия**

**Сложение и вычитание двоично-десятичных данных (4-значных)**

**B + Сложение двоично-десятичных данных (4-значных)**

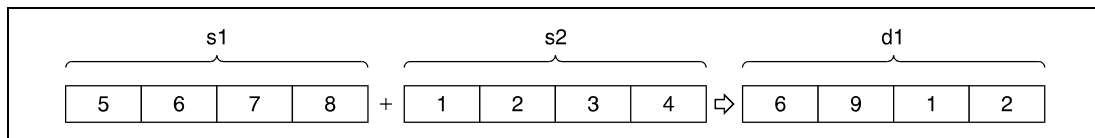
- 1-й вариант:

Указанные в s двоично-десятичные данные суммируются с двоично-десятичными данными в d. Результат сложения сохраняется в d.



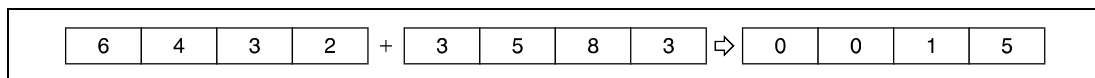
- 2-й вариант:

Указанные в s1 двоично-десятичные данные суммируются с двоично-десятичными данными в s2. Результат сложения выводится в операнд, указанный в d1.



Значение, указанное в s, d, s1, s2 и d1, должно представлять собой 4-значное десятичное число в диапазоне между 0 и 9999. Если числа имеют менее 4 знаков, первые разряды заполняются нулями (например: 12 = 0012).

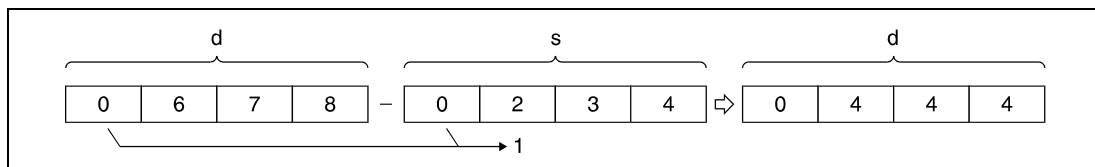
Флаг переноса не устанавливается даже в том случае, если результат сложения превышает 9999. При этом старшие биты игнорируются (переполнение).



**B - Вычитание двоично-десятичных данных (4-значных)**

- 1-й вариант:

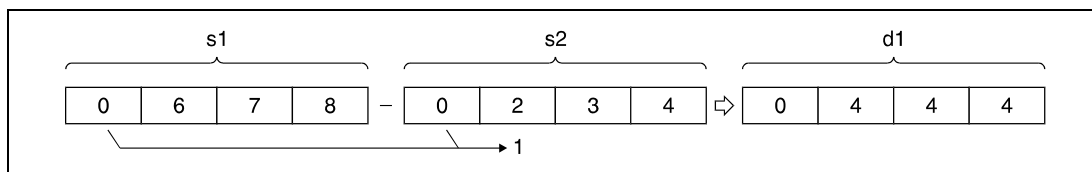
Указанные в s двоично-десятичные данные вычитаются из двоично-десятичных данных в d. Разность сохраняется в d.



<sup>1</sup> Пустые разряды заполняются нулями.

- 2-й вариант:

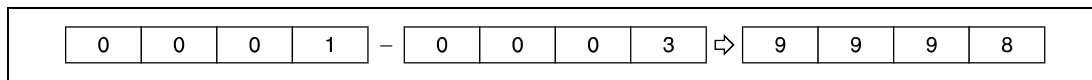
Указанные в s2 двоично-десятичные данные вычитаются из двоично-десятичных данных в s1. Результат выводится в операнд, указанный в d1.



<sup>1</sup> Пустые разряды заполняются нулями.

Значение, указанное в s, d, s1, s2 и d1, должно представлять собой 4-значное десятичное число в диапазоне между 0 и 9999.

Если разность отрицательная, то уменьшаемое уменьшается на количество шагов, зависящее от вычитаемого, и флаг переноса не устанавливается.



Программу следует составить так, чтобы ее дальнейшая работа была предусмотрена и для положительного, и для отрицательного результата.

#### Источники ошибок

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В качестве s, d, s1, s2 или d1 указано значение вне пределов 0...9999. (серии "Q"/"System Q" = код ошибки 4100).

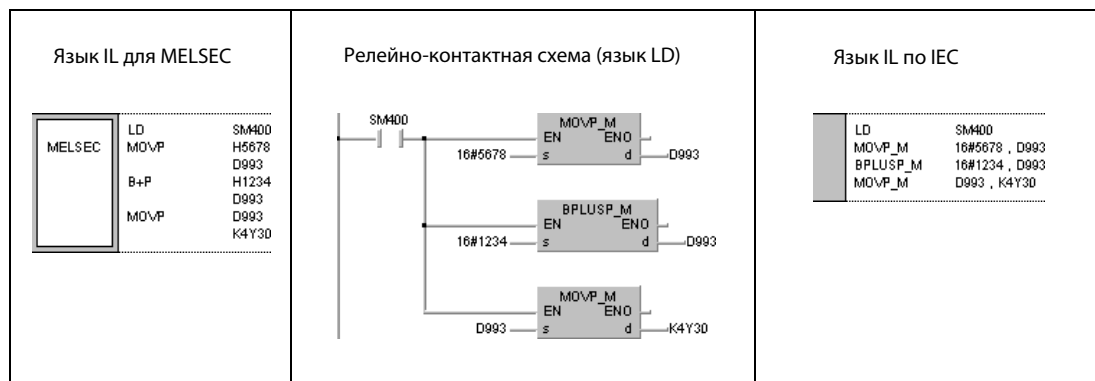
**Пример 1** B + P (s, d)

Следующая программа суммирует двоично-десятичные данные 5678 и 1234, сохраняет результат по адресу D993 и выводит его на выходы Y30...Y3F.

Для этого в первой строке программы при положительном фронте сигнала SM400 значение 5678 сохраняется в D993.

В следующем шаге к двоично-десятичному значению в D993 прибавляется двоично-десятичное значение 1234.

В последнем шаге программы команда MOV выводит результат D993 на операнды с Y30 по Y3F.



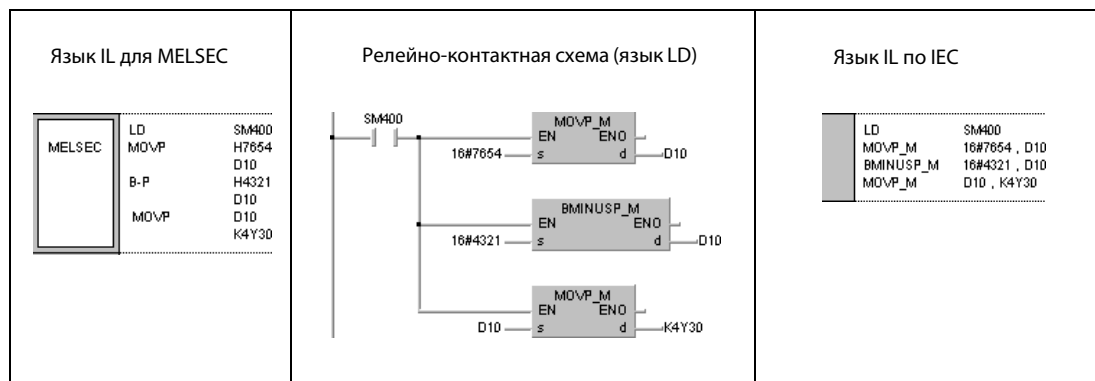
**Пример 2** B - P (s, d)

Следующая программа вычитает двоично-десятичное значение 4321 из 7654, передает результат в D10 и выводит его на выходы Y30...Y3F.

Для этого в первой строке программы при положительном фронте сигнала SM400 значение 7654 сохраняется в D10.

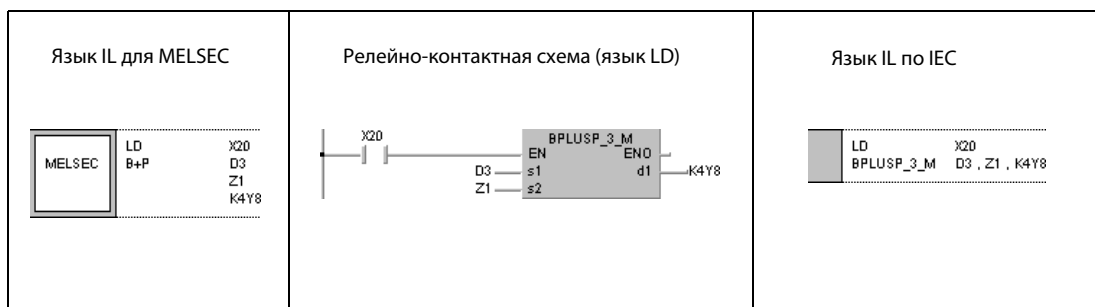
В следующем шаге из двоично-десятичного значения в D10 вычитается двоично-десятичное значение 4321.

В последнем шаге программы команда MOV выводит результат D10 на выходы с Y30 по Y3F.



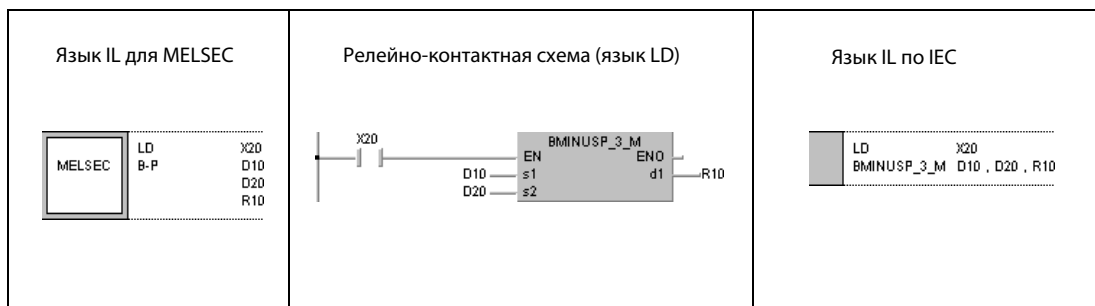
**Пример 3** B + P (s1, s2, d1)

Следующая программа при положительном фронте сигнала X20 суммирует двоично-десятичные данные в D3 и двоично-десятичные данные в Z1. Результат выводится на выходы с Y8 по Y17.



**Пример 4** B - P (s1, s2, d1)

Следующая программа при положительном фронте сигнала X20 вычитает двоично-десятичные данные в D20 из двоично-десятичных данных в D10. Результат сохраняется в R10.



6.2.6 DB +, DB + P, DB -, DB - P

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

	Операнды															Указа-тели	Уро-вень	Длина блока	Шагов	Индекс	Флаг пере-носа	Флаг ошиб-ки
	битовые						словные (16 бит)						Константы									
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z							
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					
d1		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					

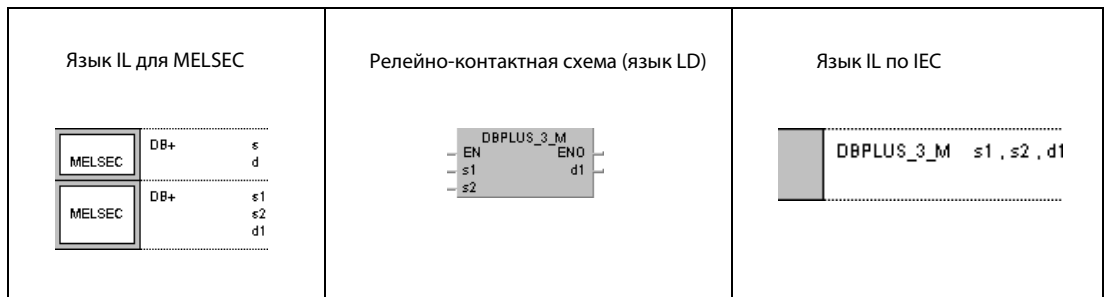
<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

<sup>2</sup> только для процессоров AnA, AnAS и AnU

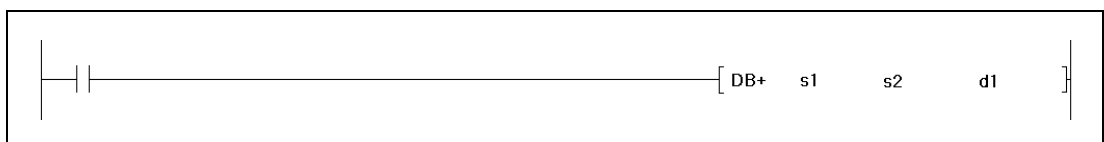
Операнды MELSEC Q

	Операнды							Флаг ошибки	Шагов		
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn			Константы K, H (16#)	Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		
s1	●	●	●	●	●	●	●	●	—	SM0	4
s2	●	●	●	●	●	●	●	●	—		
d1	●	●	●	●	●	●	●	—	—		

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
s	Слагаемые или вычитаемые данные либо первый адрес, начиная с которого сохранены эти данные.	BCD, 8 знаков
d	Слагаемые или уменьшаемые данные либо первый адрес, начиная с которого сохранены эти данные, а также первый адрес, начиная с которого сохраняется сумма или разность.	
s1	Слагаемые или уменьшаемые данные либо первый адрес, начиная с которого сохранены эти данные.	
s2	Слагаемые или вычитаемые данные либо первый адрес, начиная с которого сохранены эти данные.	
d1	Первый адрес, начиная с которого сохраняется сумма или разность.	



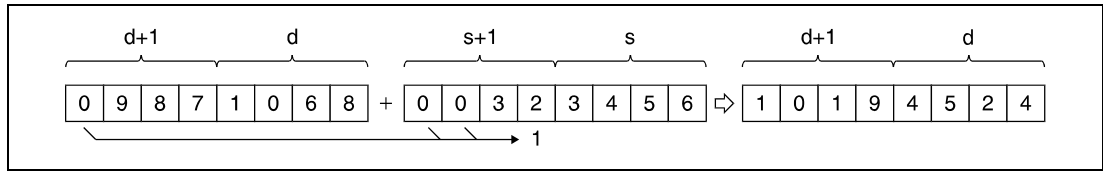
**Принцип действия**

**Сложение и вычитание двоично-десятичных данных (8-значных)**

**DB + Сложение двоично-десятичных данных (8-значных)**

● 1-й вариант:

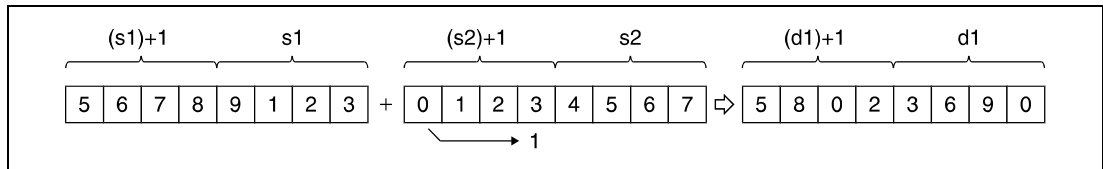
Указанные в s двоично-десятичные данные суммируются с двоично-десятичными данными в d. Результат сложения сохраняется в d.



<sup>1</sup> Пустые разряды заполняются нулями.

● 2-й вариант:

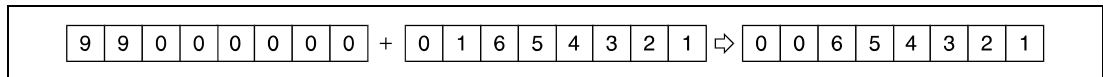
Указанные в s1 двоично-десятичные данные суммируются с двоично-десятичными данными в s2. Результат сложения выводится в операнд, указанный в d1.



<sup>1</sup> Пустые разряды заполняются нулями.

Значение, указанное в s, d, s1, s2 и d1, должно представлять собой 8-значное десятичное число в диапазоне между 0 и 99999999. Если число имеет меньше 8 знаков, первые разряды заполняются нулями (например, 12345 = 00012345).

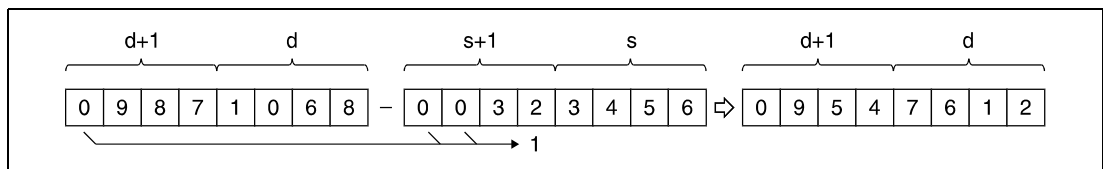
Флаг переноса не устанавливается даже в том случае, если результат сложения превышает 99999999. Старшие биты игнорируются (переполнение).



**DB - Вычитание двоично-десятичных данных (8-значных)**

● 1-й вариант:

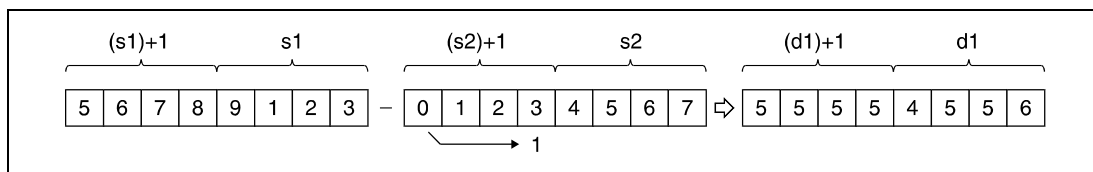
Указанные в s двоично-десятичные данные вычитаются из двоично-десятичных данных в d. Разность сохраняется в d.



<sup>1</sup> Пустые разряды заполняются нулями.

● 2-й вариант:

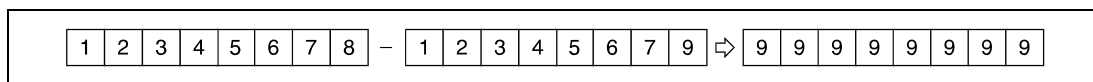
Указанные в s2 двоично-десятичные данные вычитаются из двоично-десятичных данных в s1. Результат выводится в операнд, указанный в d1.



<sup>1</sup> Пустые разряды заполняются нулями.

Значение, указанное в s, d, s1, s2 и d1, должно представлять собой 8-значное десятичное число в диапазоне между 0 и 99999999.

Если разность отрицательная, то уменьшаемое уменьшается на количество шагов, зависящее от вычитаемого, и флаг переноса не устанавливается.



Программу следует составить так, чтобы ее дальнейшая работа была предусмотрена и для положительного, и для отрицательного результата.

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В качестве s, d, s1, s2 или d1 указано значение вне пределов 0...99999999 (серии "Q"/"System Q" = код ошибки 4100).

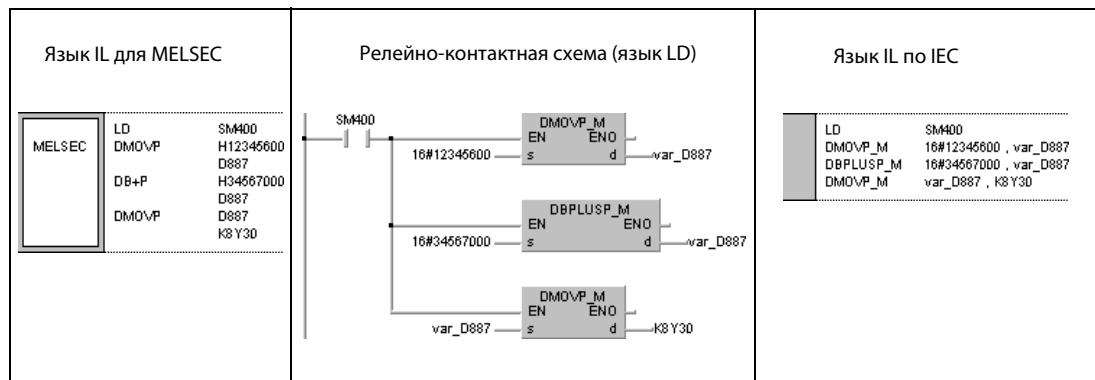
**Пример 1** DB + P (s, d)

Следующая программа суммирует двоично-десятичные данные 12345600 и 34567000, сохраняет результат по адресу D887 и D888 и выводит его на выходы с Y30 по Y4F.

В первой строке программы при положительном фронте сигнала SM400 значение 12345600 записывается в D887 и D888.

В следующем шаге двоично-десятичное значение 34567000 прибавляется к двоично-десятичному значению в D887 и D888.

В последнем шаге программы команда DMOV P выводит результат D887 и D888 на выходы Y30...Y4F.



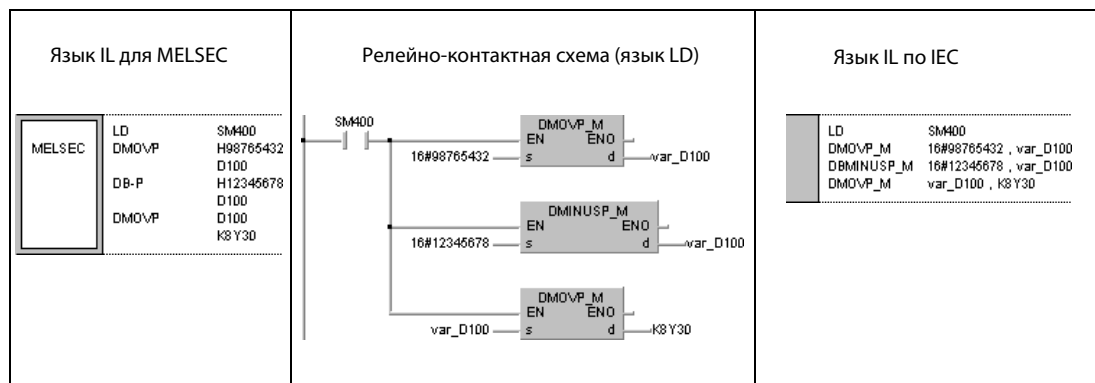
**Пример 2** DB - P (s, d)

Следующая программа вычитает двоично-десятичные данные 12345678 из 98765432, сохраняет результат по адресу D100 и D101 и выводит его на выходы с Y30 по Y4F.

Для этого в первой строке программы при положительном фронте сигнала SM400 значение 98765432 записывается в D100 и D101.

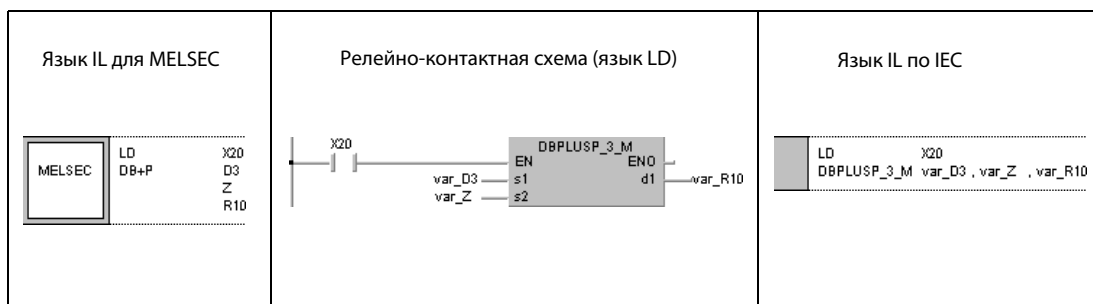
В следующем шаге из двоично-десятичного значения в D100 и D101 вычитается двоично-десятичное значение 12345678.

В последнем шаге программы команда DMOV P выводит результат D100 и D101 на выходы Y30...Y4F.



**Пример 3** DB + P (s1, s2, d1)

Следующая программа при положительном фронте сигнала X20 суммирует двоично-десятичные данные в D3 и D4 с двоично-десятичными данными в Z и V. Результат сохраняется в R10 и R11.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

**6.2.7 В х, В х Р, В/, В/Р**

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

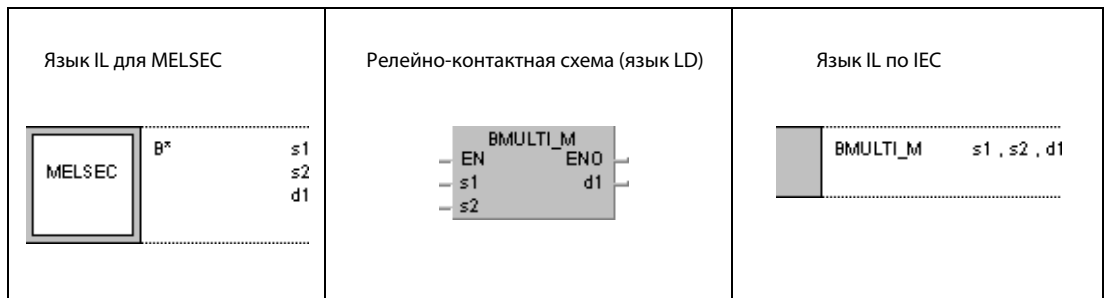
	Операнды																Указа- тели	Уро- вень	Длина блока	Шагов	Индекс	Флаг пере- носа	Флаг ошибки			
	битовые						словные (16 бит)						Константы		M9012	M9010 M9011										
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1										Z	V	K
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	9			
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K8	1	●		●
d1		●	●	●	●	●	●	●	●	●	●	●														

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

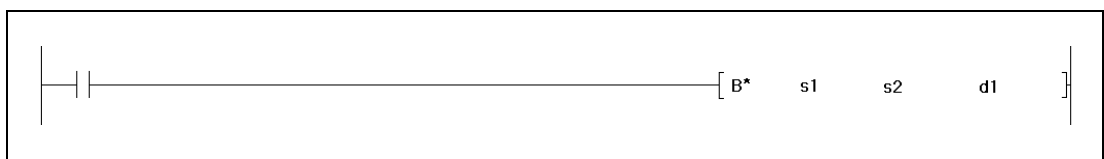
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	●	●	●	●	●	●	●	●	—	SM0	4
s2	●	●	●	●	●	●	●	●	—		
d1	●	●	●	●	●	●	●	—	—		

**GX IEC Developer**



**GX Developer**

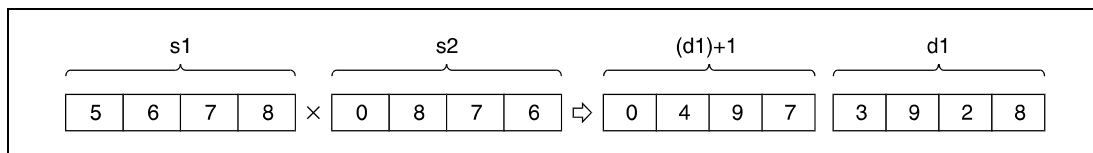


**Переменные**

Операнд	Значение	Тип данных	
		MELSEC	IEC
s1	Данные множителя или делимого либо первый адрес, начиная с которого сохранены эти данные.	BCD, 4 знака	WORD
s2	Данные множителя или делителя либо первый адрес, начиная с которого сохранены эти данные.	BCD, 4 знака	WORD
d1	Первый адрес, начиная с которого сохраняется произведение или частное.	BCD, 8 знаков	2 массива данных типа WORD

**Принцип действия****Умножение и деление двоично-десятичных данных (4-значных)****Вх      Умножение двоично-десятичных данных (4-значных)**

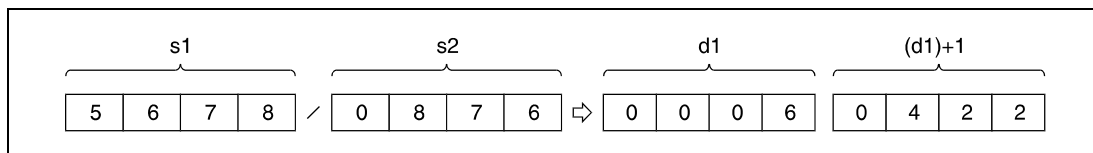
Указанные в s1 двоично-десятичные данные умножаются на двоично-десятичные данные в s2 и произведение сохраняется в d1.



Указанное в s1 и s2 значение должно представлять собой 4-значное десятичное число в диапазоне между 0 и 9999.

**В/      Деление двоично-десятичных данных (4-значных)**

Указанные в s1 двоично-десятичные данные делятся на двоично-десятичные данные в s2 и результат деления сохраняется в d1.



Результат деления сохраняется в двух массивах данных типа WORD (BCD, 8 знаков). При этом результат разделяется на частное и остаток деления.

Частное (4-значное число BCD) сохраняется в младших элементах массива. Остаток (4-значное число BCD) сохраняется в старших элементах массива. В случае битовых операндов остаток результата деления не сохраняется.

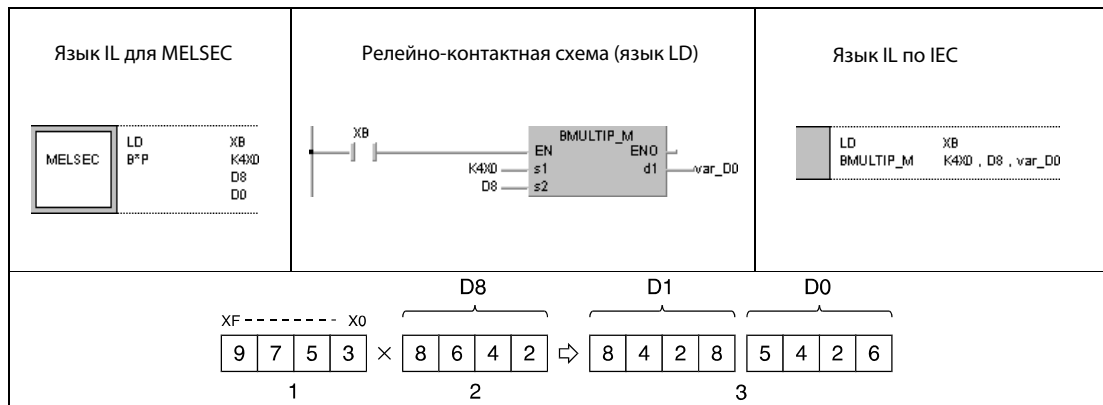
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В качестве данных в s1 или s2 указано значение вне пределов 0...9999.
- Делитель s2 равен 0 (серии "Q"/"System Q" = код ошибки 4100).

**Пример 1** В х Р

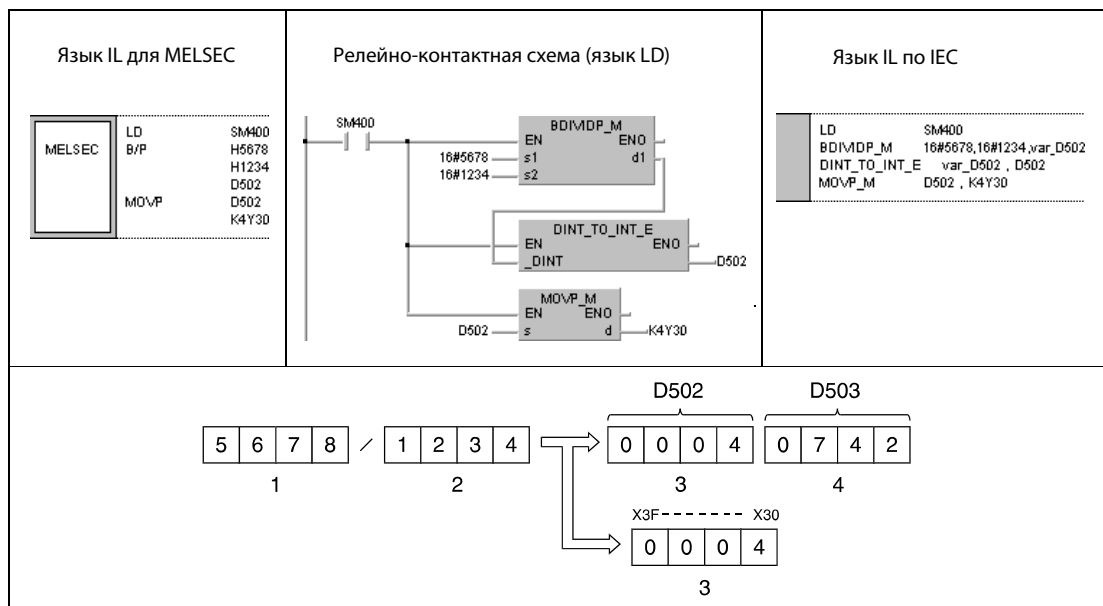
Следующая программа при положительном фронте сигнала XВ умножает двоично-десятичные данные из X0...XF на двоично-десятичные данные из D8. Результат сохраняется в D0 и D1.



- 1 умножаемое
- 2 множитель
- 3 произведение

**Пример 2** В/Р

В следующем примере при положительном фронте сигнала SM400 двоично-десятичные данные 5678 делятся на 1234. Результат сохраняется в D502, а остаток – в D503. В последнем шаге частное (D502) выводится на выходы с Y30 по Y3F.



- 1 делимое
- 2 делитель
- 3 частное
- 4 остаток

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию можно найти в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

6.2.8 DB x, DB x P, DB/, DB/P

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

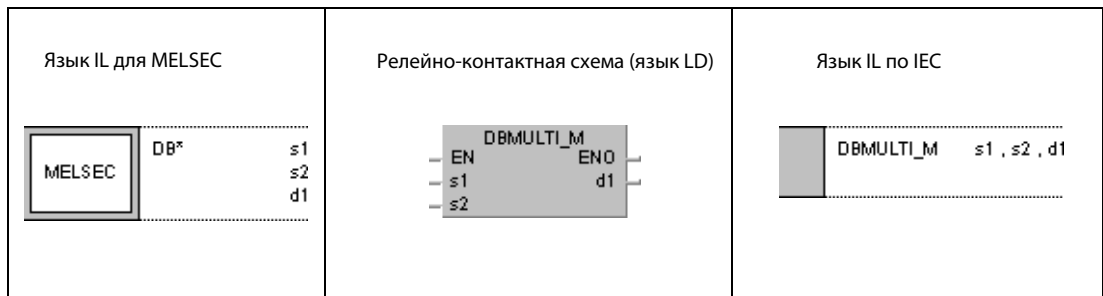
	Операнды																Указа-тели	Уро-вень	Длина блока	Шагов	Индекс	Флаг пере-носа	Флаг оши-бки		
	битовые						словные (16 бит)						Константы		M9012	M9010 M9011									
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1										Z	V
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ KB	11	●	●
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								
d1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

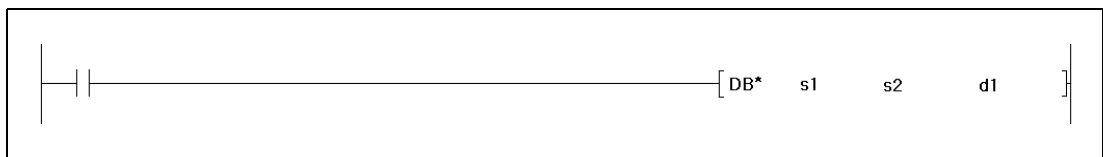
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	●	●	●	●	●	●	●	●	—	SM0	4
s2	●	●	●	●	●	●	●	—			
d1	●	●	●	—	—	—	—	—			

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
s1	Данные множителя или делимого либо первый адрес, начиная с которого сохранены эти данные.	BCD, 8 знаков
s2	Данные множителя или делителя либо первый адрес, начиная с которого сохранены эти данные.	BCD, 8 знаков
d1	Первый адрес, начиная с которого сохраняется произведение или частное.	BCD, 16 знаков

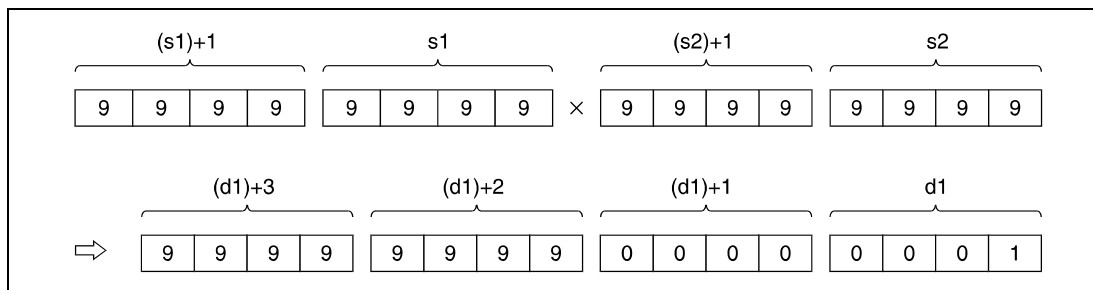


**Принцип действия**

**Умножение и деление двоично-десятичных данных (8-значных)**

**DB x Умножение двоично-десятичных данных (8-значных)**

Указанные в s1 двоично-десятичные данные умножаются на двоично-десятичные данные в s2 и произведение сохраняется в d1.



Если d1 представляет собой битовый операнд, то возможна обработка только последних 8 знаков (младших 32 битов).

Пример:

K1: 4 младших бита (от b0 до b3)

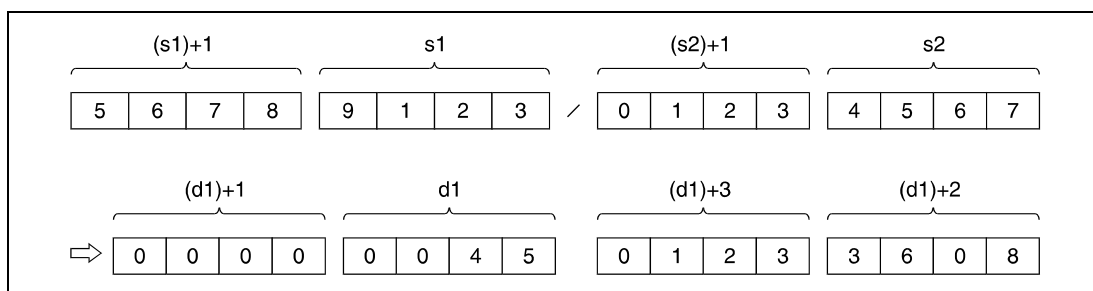
K4: 16 младших битов (от b0 до b15)

K8: 32 бита (от b0 до b31)

Указанное в s1 и s2 значение должно представлять собой 8-значное десятичное число в диапазоне между 0 и 99999999. Если число имеет меньше 8 знаков, первые разряды заполняются нулями (например: 12345 = 00012345).

**DB/ Деление двоично-десятичных данных (8-значных)**

Указанные в s1 двоично-десятичные данные делятся на двоично-десятичные данные в s2 и результат деления сохраняется в d1.



Результат деления сохраняется в виде массива из двух 32-битных слов. При этом он разделяется на частное и остаток деления.

Частное (8-значное число BCD) сохраняется в 32 младших битах. Остаток (8-значное число BCD) сохраняется в 32 старших битах. В случае битовых операндов остаток результата деления не сохраняется.

**Источники ошибок**

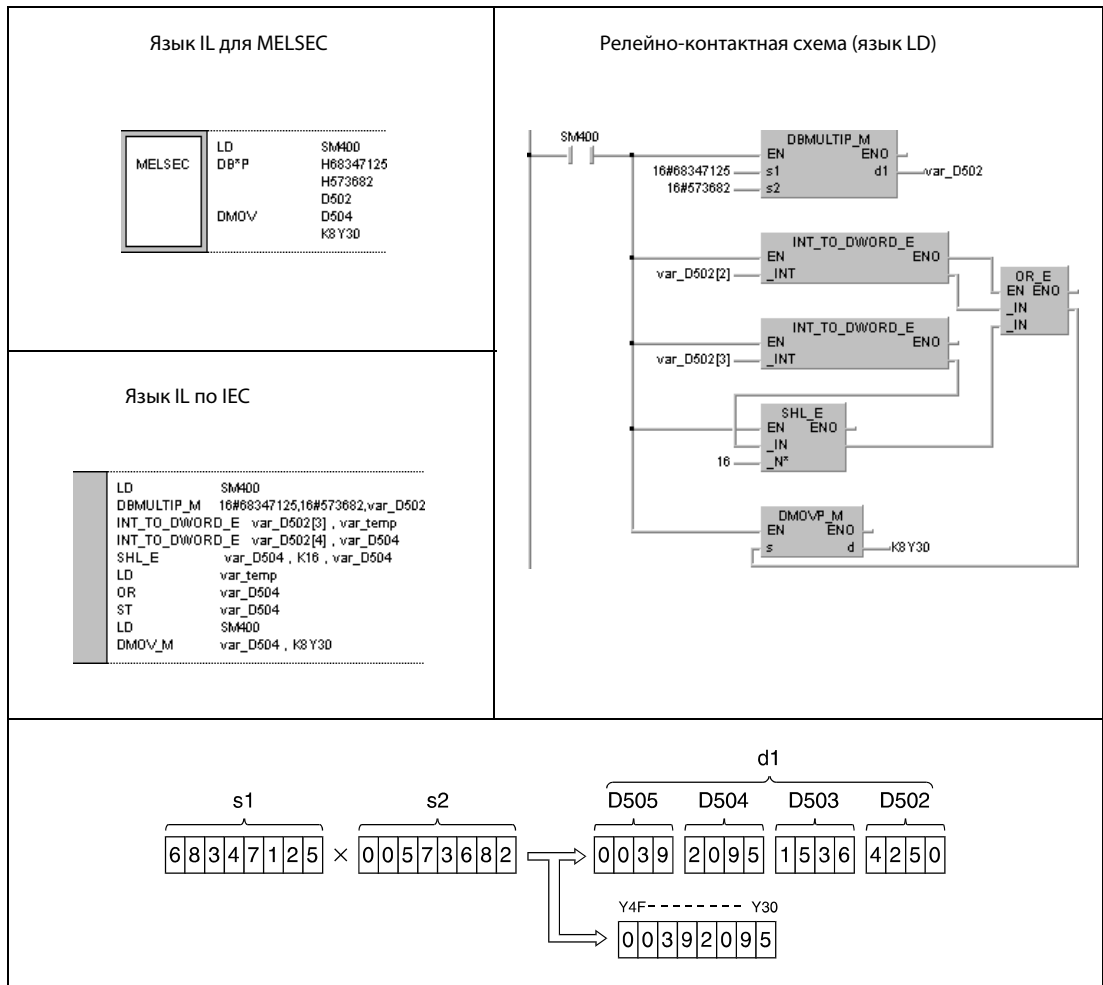
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В качестве данных в s1 или s2 указано значение вне пределов 0...99999999.
- Делитель s2 равен 0 (серия "Q" = код ошибки 4100).

**Пример 1**

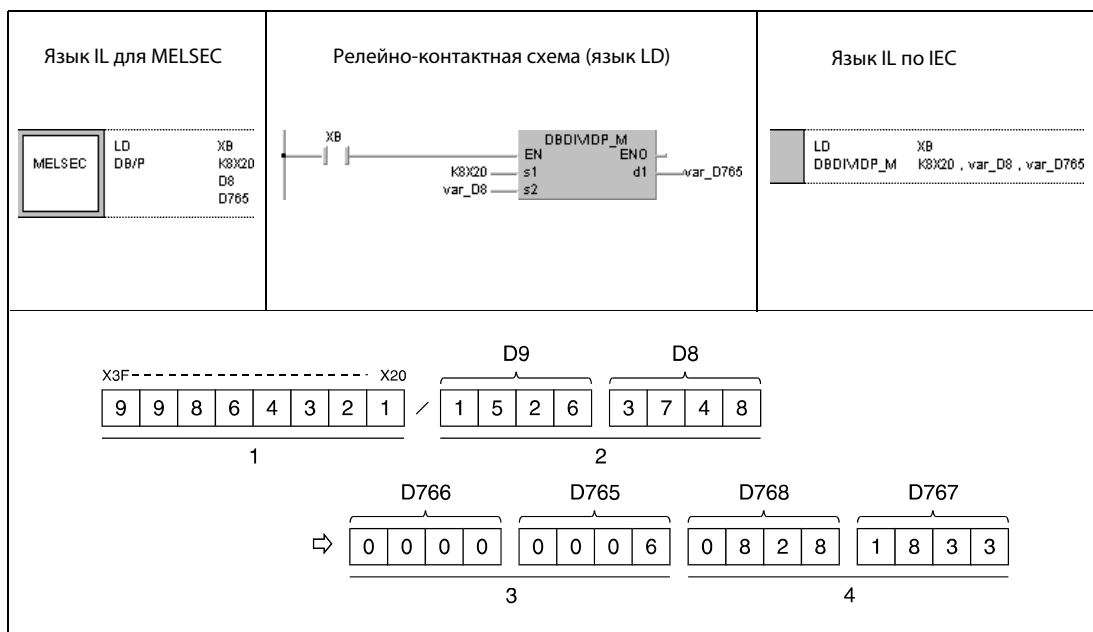
DB x P

Следующая программа при положительном фронте сигнала SM400 умножает двоично-десятичные данные 68347125 и 573682 и сохраняет результат по адресу D502...D505. В следующем шаге программы 8 старших знаков результата (D504, D505) выводятся на выходы с Y30 по Y4F.



**Пример 2** DB/P

Следующая программа при положительном фронте сигнала XВ делит двоично-десятичные данные X20...X3F на содержимое регистров D8 и D9. Результат сохраняется в D765...D768.



- <sup>1</sup> делимое
- <sup>2</sup> делитель
- <sup>3</sup> частное
- <sup>4</sup> остаток

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

### 6.2.9 E +, E + P, E -, E - P

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

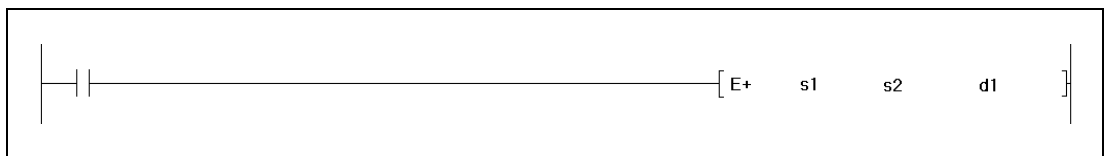
Операнды MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы E	Иные		
	битовые	словные		битовые	словные						
s	—	●	●	—	●	●	—	●	—	SM0	3
d	—	●	●	—	●	●	—	—	—		
s1	—	●	●	—	●	●	—	●	—	SM0	4
s2	—	●	●	—	●	●	—	●	—		
d1	—	●	●	—	●	●	—	—	—		

GX IEC Developer



GX Developer



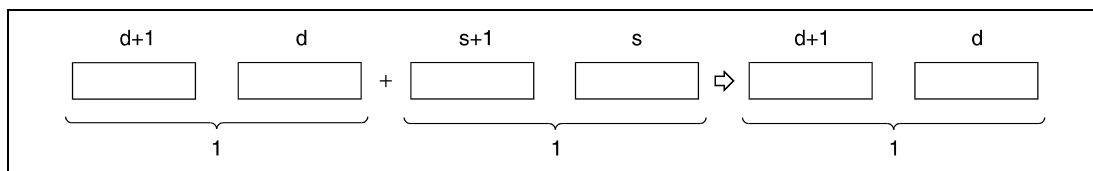
Переменные

Операнд	Значение	Тип данных
s	Слагаемые или вычитаемые данные либо первый адрес, начиная с которого сохранены эти данные.	REAL
d	Слагаемые или уменьшаемые данные либо первый адрес, начиная с которого сохранены эти данные, а также первый адрес, начиная с которого сохраняется сумма или разность.	
s1	Слагаемые или уменьшаемые данные либо первый адрес, начиная с которого сохранены эти данные.	
s2	Слагаемые или вычитаемые данные либо первый адрес, начиная с которого сохранены эти данные.	
d1	Первый адрес, начиная с которого сохраняется сумма или разность.	

**Принцип действия****Сложение и вычитание чисел с плавающей запятой****E + Команда сложения чисел с плавающей запятой**

## ● 1-й вариант:

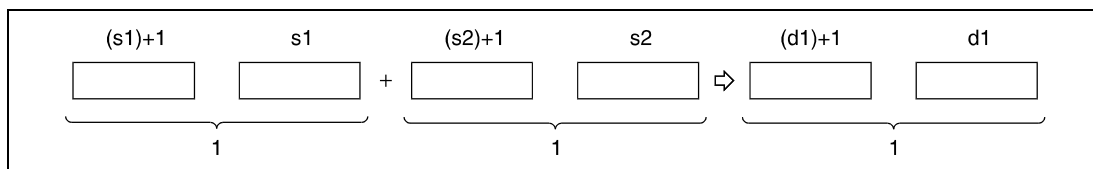
Число с плавающей запятой, указанное в s, прибавляется к числу с плавающей запятой в d. Результат сложения сохраняется в d.



<sup>1</sup> число с плавающей запятой (данные типа REAL)

## ● 2-й вариант:

Указанное в s1 число с плавающей запятой прибавляется к числу с плавающей запятой в s2. Результат сложения выводится в операнд, указанный в d1.



<sup>1</sup> число с плавающей запятой (данные типа REAL)

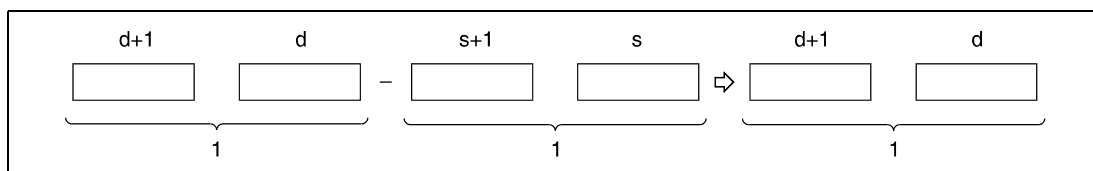
Указанное в s, d, s1, s2 и d1 или сохраняемое значение должно быть либо равным нулю, либо представлять собой число с плавающей запятой в следующих пределах:

$$\pm 2^{-127} \leq \text{число с плавающей запятой (s, d, s1, s2, d1)} < \pm 2^{129}$$

**E - Команда вычитания чисел с плавающей запятой**

## ● 1-й вариант:

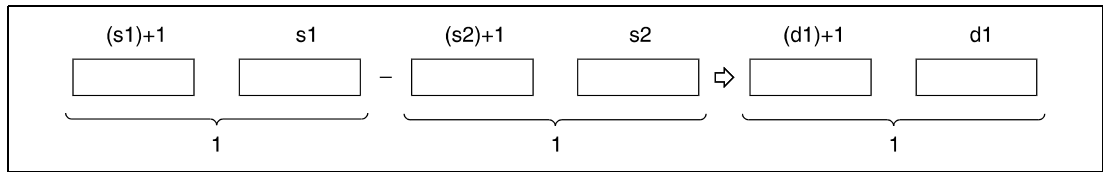
Число с плавающей запятой, указанное в s, вычитается из числа с плавающей запятой в d. Разность сохраняется в d.



<sup>1</sup> число с плавающей запятой (данные типа REAL)

● 2-й вариант:

Указанное в s2 число с плавающей запятой вычитается из числа с плавающей запятой в s1. Разность выводится в операнд, указанный в d1.



<sup>1</sup> число с плавающей запятой (данные типа REAL)

Указанное в s, d, s1, s2 и d1 или сохраняемое значение должно быть либо равным нулю, либо представлять собой число с плавающей запятой в следующих пределах:

$$\pm 2^{-127} \leq \text{число с плавающей запятой (s, d, s1, s2, d1)} < \pm 2^{129}$$

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

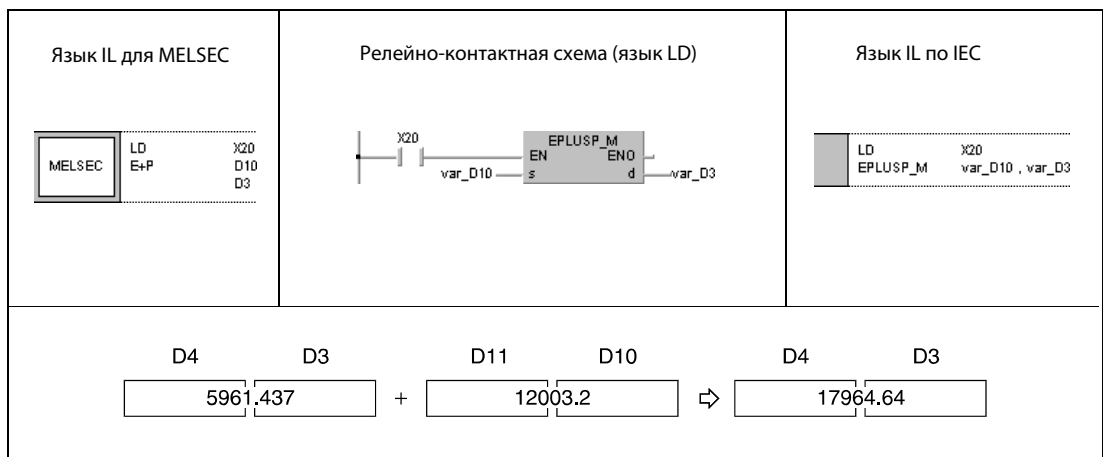
- Указанные в s, d, s1, s2 и d1 числа с плавающей запятой или результаты сложения или вычитания не равны нулю и находятся вне следующих пределов (код ошибки 4100):

$$\pm 2^{-127} \leq \text{число с плавающей запятой/результат операции (s, d, s1, s2, d1)} < \pm 2^{129}$$

**Пример 1**

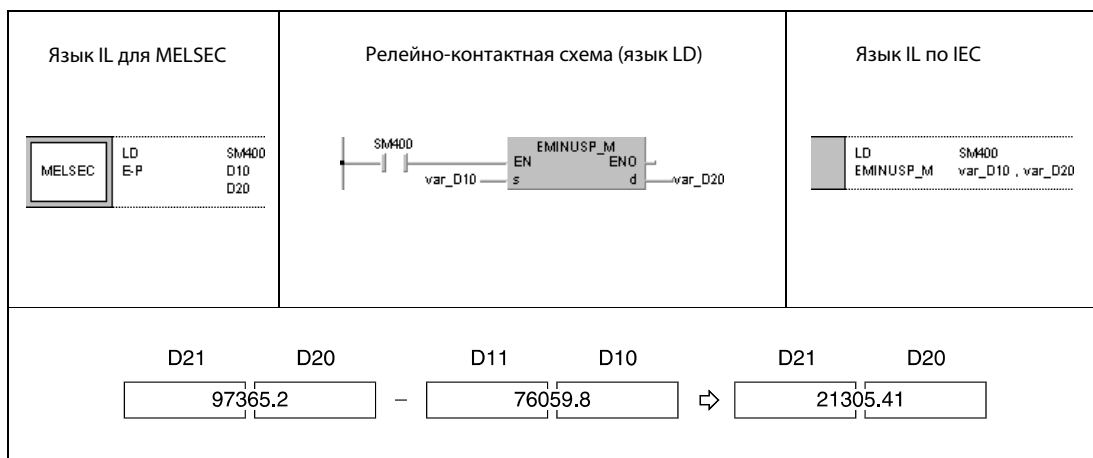
E + P (s, d)

Следующая программа при положительном фронте сигнала X20 суммирует число с плавающей запятой в D3 и D4 с числом с плавающей запятой в D10 и D11. Сумма сохраняется в D3 и D4.



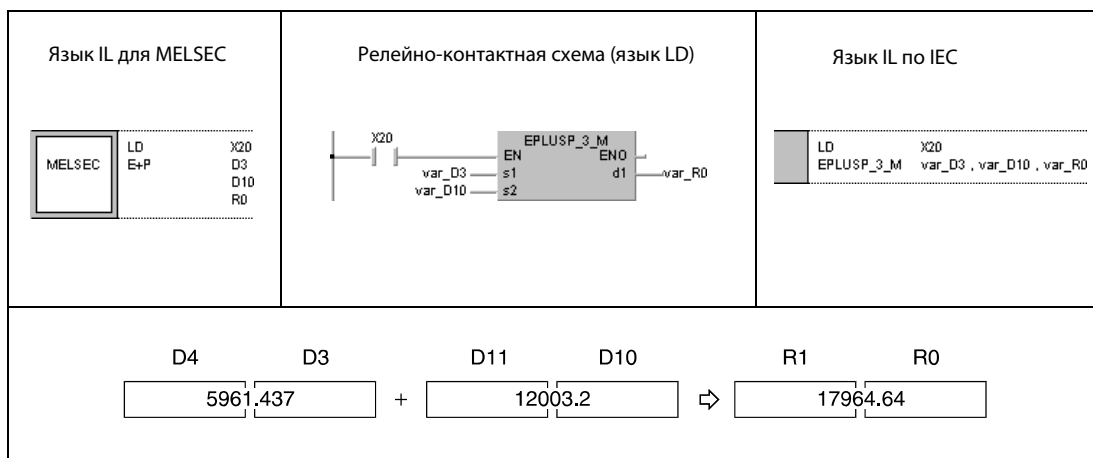
**Пример 2** E - P (s, d)

Следующая программа при положительном фронте сигнала SM400 вычитает число с плавающей запятой в D10 и D11 из числа с плавающей запятой в D20 и D21. Разность сохраняется в D20 и D21.



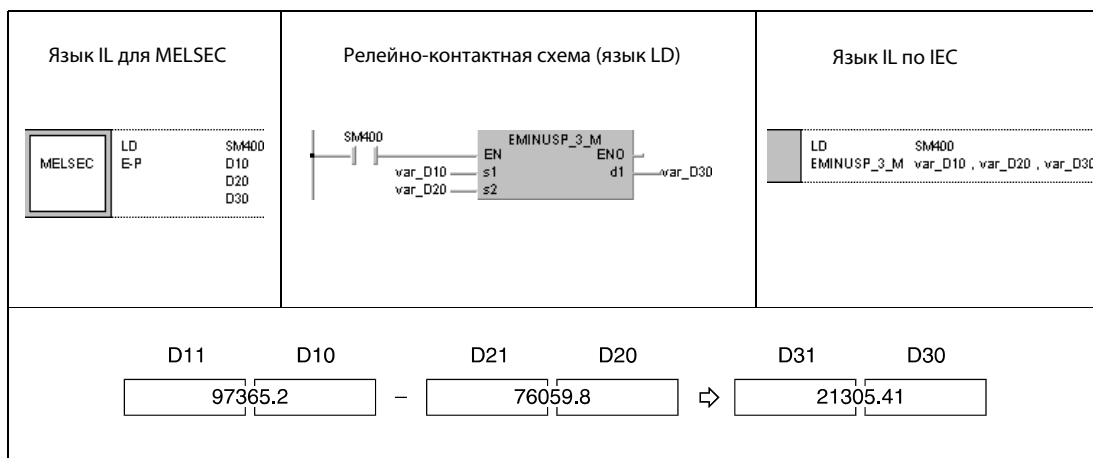
**Пример 3** E + P (s1, s2, d)

Следующая программа при положительном фронте сигнала X20 суммирует числа с плавающей запятой в D3 и D4 с числами с плавающей запятой в D10 и D11. Сумма сохраняется в R0 и R1.



**Пример 4** E - P (s1, s2, d)

Следующая программа при положительном фронте SM400 вычитает числа с плавающей запятой в D20 и D21 из чисел с плавающей запятой в D10 и D11. Результат сохраняется в D30 и D31.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.



### 6.2.10 Ех, ЕхР, Е/, Е/Р

**Процессор**

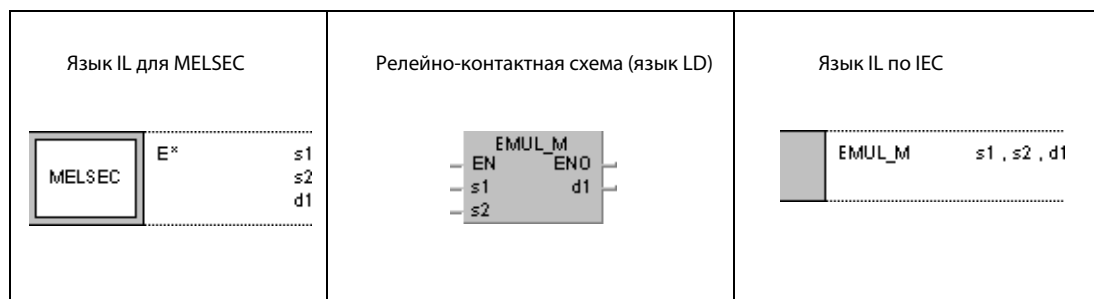
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

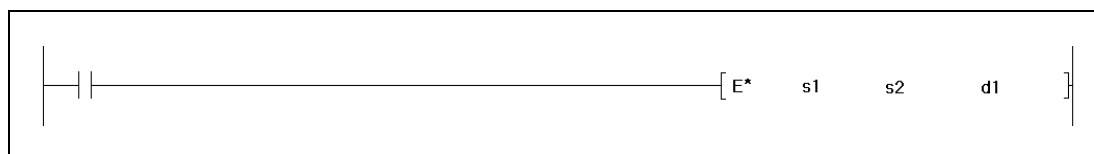
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы E	Иные U		
	битовые	словные		битовые	словные						
s1	—	●	●	—	●	●	—	●	—	SMO	4
s2	—	●	●	—	●	●	—	●	—		
d1	—	●	●	—	●	●	—	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

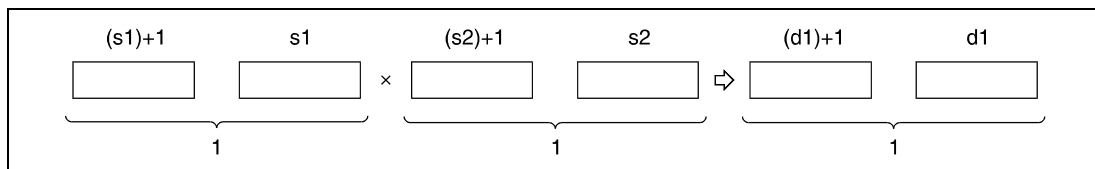
Операнд	Значение	Тип данных
s1	Данные множителя или делимого либо первый адрес, начиная с которого сохранены эти данные.	REAL
s2	Данные множителя или делителя либо первый адрес, начиная с которого сохранены эти данные.	
d1	Первый адрес, начиная с которого сохраняется произведение или частное.	

**Принцип действия**

**Умножение и деление чисел с плавающей запятой**

**Ех Команда умножения чисел с плавающей запятой**

Указанное в s1 число с плавающей запятой умножается на число с плавающей запятой в s2 и произведение сохраняется в d1.



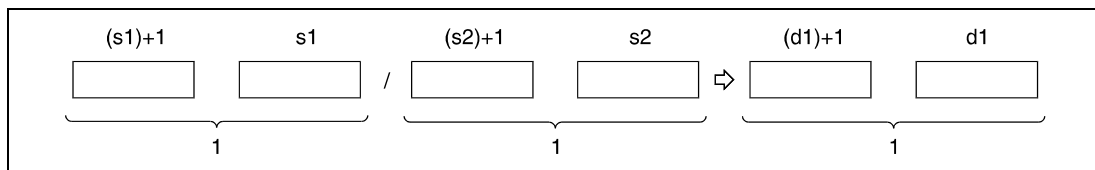
<sup>1</sup> число с плавающей запятой (данные типа REAL)

Указанное в s1, s2 и d1 или сохраняемое значение должно быть либо равным нулю, либо представлять собой число с плавающей запятой в следующих пределах:

$$\pm 2^{-127} \leq \text{число с плавающей запятой (s1, s2, d1)} < \pm 2^{129}$$

**Е/ Команда деления чисел с плавающей запятой**

Указанное в s1 число с плавающей запятой делится на число с плавающей запятой в s2 и результат деления сохраняется в d1.



<sup>1</sup> число с плавающей запятой (данные типа REAL)

Указанное в s1, s2 и d1 или сохраняемое значение должно быть либо равным нулю, либо представлять собой число с плавающей запятой в следующих пределах:

$$\pm 2^{-127} \leq \text{число с плавающей запятой (s1, s2, d1)} < \pm 2^{129}$$

**Источники ошибок**

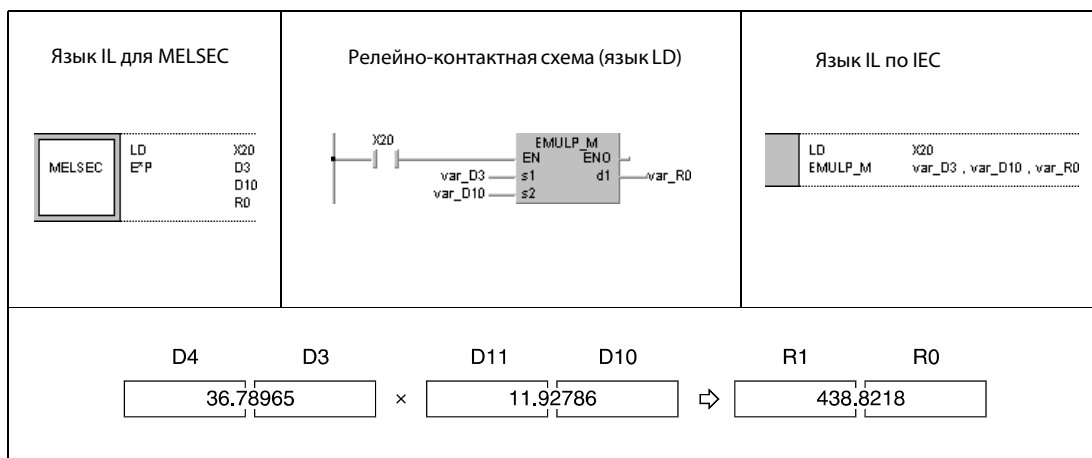
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанные в s1, s2 и d1 числа с плавающей запятой или результаты умножения или деления не равны нулю и находятся вне следующих пределов (код ошибки 4100):  

$$\pm 2^{-127} \leq \text{число с плавающей запятой/результат операции (s1, s2, d1)} < \pm 2^{129}$$
- Делитель s2 равен 0 (код ошибки 4100).

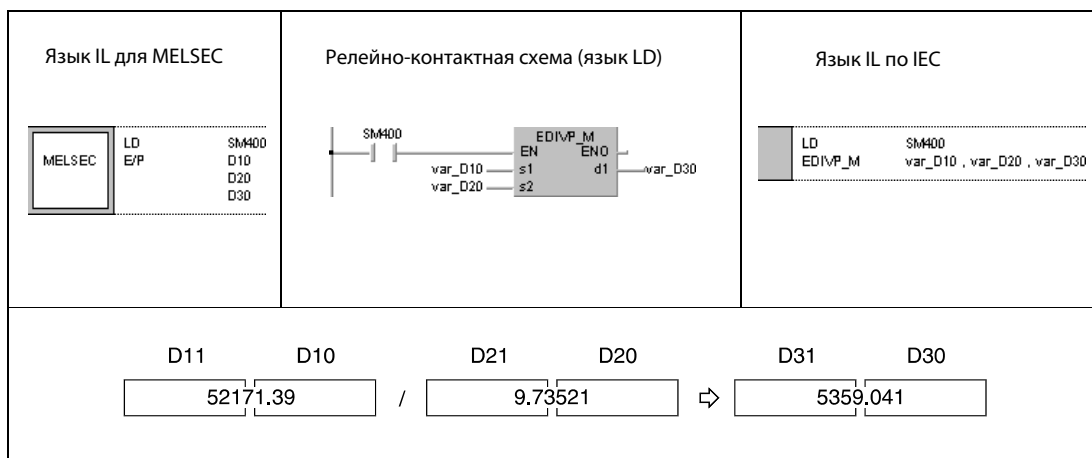
**Пример 1** ЕхР

Следующая программа при положительном фронте сигнала X20 умножает число с плавающей запятой в D3 и D4 на число с плавающей запятой в D10 и D11. Произведение сохраняется в R0 и R1.



**Пример 2** Е/Р

Следующая программа при положительном фронте сигнала SM400 делит число с плавающей запятой в D10 и D11 на число с плавающей запятой в D20 и D21. Результат деления сохраняется в D30 и D31.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

6.2.11 БК +, БК + P, БК -, БК - P

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

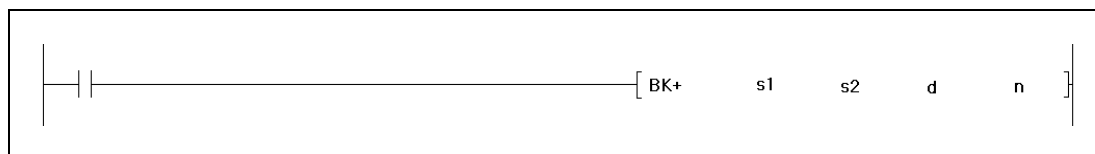
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□□		Спец. модули U□\G□□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	SMO	5	
s2	—	●	●	—	—	—	—	●			
d	—	●	●	—	—	—	—	—			
n	●	●	●	●	●	●	●	—			

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
s1	Слагаемые или уменьшаемые данные либо первый адрес, начиная с которого сохранены эти данные.	BIN, 16 бит
s2	Первый адрес, начиная с которого сохранены слагаемые или вычитаемые данные.	
d	Первый адрес, начиная с которого должны сохраняться данные результата.	
n	Количество блоков данных, в отношении которых выполняется операция.	

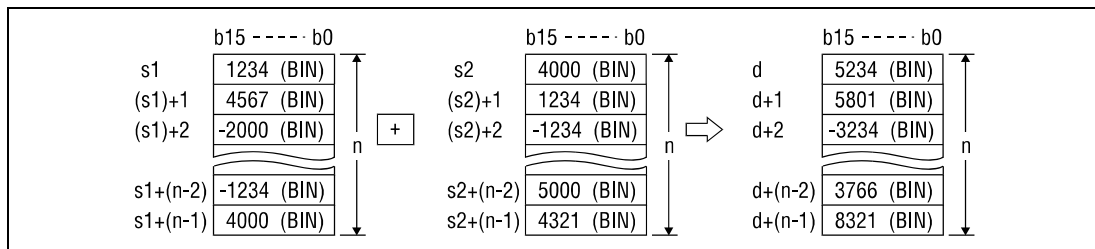
**Принцип действия**

**Поблочное сложение и вычитание двоичных данных**

**ВК + Команда сложения блоков двоичных данных**

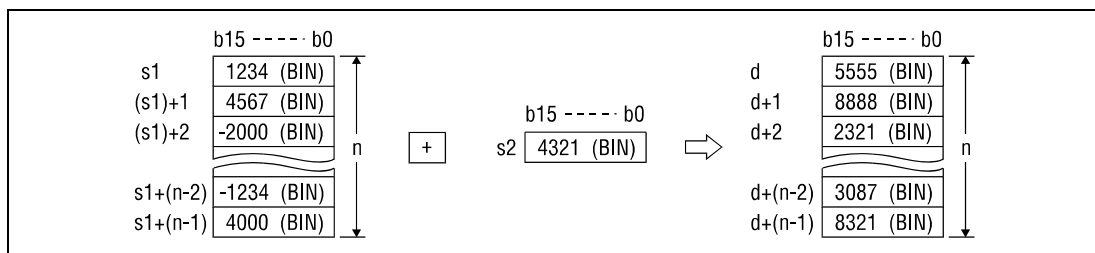
Команда сложения блоков двоичных данных состоит из самой команды, суммируемых данных  $s1$  и  $s2$ , обозначения цели  $d$ , в которой сохраняются результаты, и количества  $n$  суммируемых блоков данных.

Команда суммирует  $n$ -ый 16-битный блок данных  $s1$  с  $n$ -ым 16-битным блоком данных  $s2$ , начиная с младшего 16-битного блока. Результат каждого сложения блоков сохраняется в  $d$ .



Операции сложения выполняются единицами по 16 битов.

Сохраненная в  $s2$  константа должна представлять собой 16-битное двоичное число в диапазоне между  $-32768$  и  $32767$ .



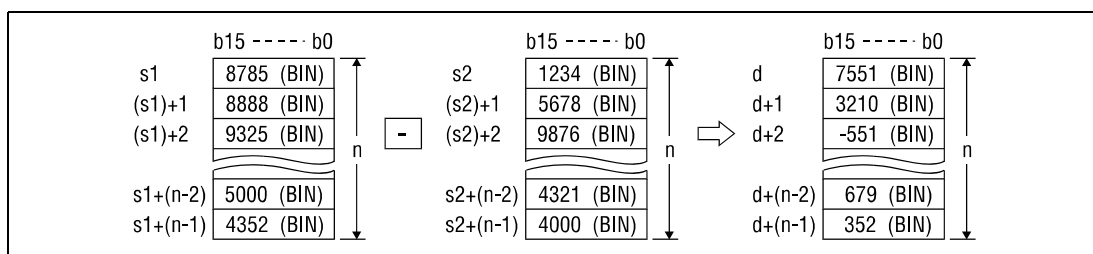
Самый старший бит блока устанавливает арифметический знак значений данных в соответствующем блоке  $s1$ ,  $s2$  или  $d$ . Если они положительные, то этот бит = 0. Если они отрицательные, этот бит = 1.

При занижении самого младшего бита блока или превышении самого старшего бита блока флаг переноса не устанавливается.

**ВК - Команда вычитания блоков двоичных данных**

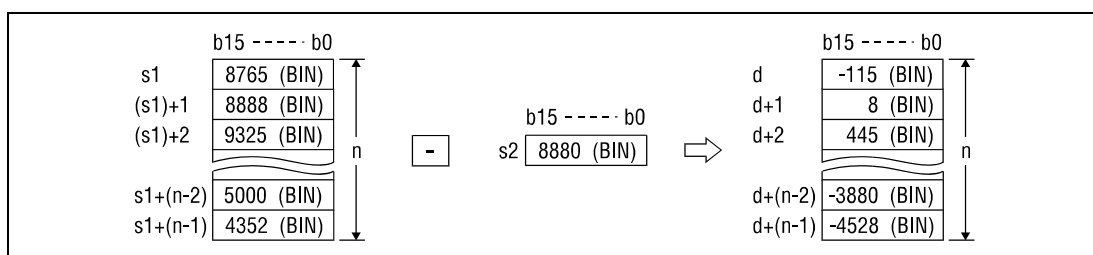
Команда вычитания блоков двоичных данных состоит из самой команды, данных s1 и s2, для которых определяется разность, обозначения цели d, в которой сохраняются результаты, и количества n вычитаемых блоков данных.

Команда вычитает n-ый 16-битный блок данных s2 из n-ного 16-битного блока данных s1, начиная с младшего 16-битного блока. Результат каждого вычитания блока сохраняется в d.



Операции вычитания выполняются единицами по 16 битов.

Сохраненная в s2 константа должна представлять собой 16-битное двоичное число в диапазоне между -32768 и 32767.



Самый старший бит блока устанавливает арифметический знак значений данных в соответствующем блоке s1, s2 или d. Если они положительные, то этот бит = 0. Если они отрицательные, этот бит = 1.

При занижении самого младшего бита блока или превышении самого старшего бита блока флаг переноса не устанавливается.

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Количество блоков данных в s1, s2 или d превышает допустимое.
- Область данных s1 перекрывается с областями s2 или d.

**Пример 1** БК + P

Следующая программа при положительном фронте сигнала X20 суммирует блоки данных, начиная с D100, с блоками данных, начиная с R0, и сохраняет результирующие блоки, начиная с D200. Количество суммируемых блоков (4) указано в D0.

Язык IL для MELSEC	Релейно-контактная схема (язык LD)	Язык IL по IEC																														
<div style="border: 1px solid black; padding: 5px; display: inline-block;">MELSEC</div> <pre> LD      X20 BK+P   D100         R0         D200         D0                     </pre>		<pre> LD      X20 BKPLUSP_M  D100 , R0 , D0 , D200                     </pre>																														
<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%; text-align: center;">b15 ----- b0</th> <th style="width: 10%;"></th> <th style="width: 33%; text-align: center;">b15 ----- b0</th> <th style="width: 10%;"></th> <th style="width: 33%; text-align: center;">b15 ----- b0</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">D100 6789 (BIN)</td> <td></td> <td style="text-align: center;">R0 1234 (BIN)</td> <td></td> <td style="text-align: center;">D200 8023 (BIN)</td> </tr> <tr> <td style="text-align: center;">D101 7821 (BIN)</td> <td style="text-align: center;">+</td> <td style="text-align: center;">R1 2032 (BIN)</td> <td style="text-align: center;">⇒</td> <td style="text-align: center;">D201 9853 (BIN)</td> </tr> <tr> <td style="text-align: center;">D102 5432 (BIN)</td> <td></td> <td style="text-align: center;">R2 -3252 (BIN)</td> <td></td> <td style="text-align: center;">D202 2180 (BIN)</td> </tr> <tr> <td style="text-align: center;">D103 3520 (BIN)</td> <td></td> <td style="text-align: center;">R3 -1000 (BIN)</td> <td></td> <td style="text-align: center;">D203 2520 (BIN)</td> </tr> <tr> <td style="text-align: center;">D0 4</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>			b15 ----- b0		b15 ----- b0		b15 ----- b0	D100 6789 (BIN)		R0 1234 (BIN)		D200 8023 (BIN)	D101 7821 (BIN)	+	R1 2032 (BIN)	⇒	D201 9853 (BIN)	D102 5432 (BIN)		R2 -3252 (BIN)		D202 2180 (BIN)	D103 3520 (BIN)		R3 -1000 (BIN)		D203 2520 (BIN)	D0 4				
b15 ----- b0		b15 ----- b0		b15 ----- b0																												
D100 6789 (BIN)		R0 1234 (BIN)		D200 8023 (BIN)																												
D101 7821 (BIN)	+	R1 2032 (BIN)	⇒	D201 9853 (BIN)																												
D102 5432 (BIN)		R2 -3252 (BIN)		D202 2180 (BIN)																												
D103 3520 (BIN)		R3 -1000 (BIN)		D203 2520 (BIN)																												
D0 4																																

**Пример 2** БК - P

Следующая программа при положительном фронте сигнала X1C вычитает константу 8765 из блоков данных, начиная с D100, и сохраняет результирующие блоки, начиная с R0. Количество блоков данных (3) указано константой K3.

Язык IL для MELSEC	Релейно-контактная схема (язык LD)	Язык IL по IEC																				
<div style="border: 1px solid black; padding: 5px; display: inline-block;">MELSEC</div> <pre> LD      X1C BK-P   D100         K3765         R0         K3                     </pre>		<pre> LD      X1C BKMINUSP_M  D100 , 8765 , 3 , R0                     </pre>																				
<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 33%; text-align: center;">b15 ----- b0</th> <th style="width: 10%;"></th> <th style="width: 33%; text-align: center;">b15 ----- b0</th> <th style="width: 10%;"></th> <th style="width: 33%; text-align: center;">b15 ----- b0</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">D100 12345 (BIN)</td> <td></td> <td style="text-align: center;">R0 3580 (BIN)</td> <td></td> <td></td> </tr> <tr> <td style="text-align: center;">D101 8701 (BIN)</td> <td style="text-align: center;">-</td> <td style="text-align: center;">8765 (BIN)</td> <td style="text-align: center;">⇒</td> <td style="text-align: center;">R1 -64 (BIN)</td> </tr> <tr> <td style="text-align: center;">D102 3502 (BIN)</td> <td></td> <td></td> <td></td> <td style="text-align: center;">R2 -5263 (BIN)</td> </tr> </tbody> </table>			b15 ----- b0		b15 ----- b0		b15 ----- b0	D100 12345 (BIN)		R0 3580 (BIN)			D101 8701 (BIN)	-	8765 (BIN)	⇒	R1 -64 (BIN)	D102 3502 (BIN)				R2 -5263 (BIN)
b15 ----- b0		b15 ----- b0		b15 ----- b0																		
D100 12345 (BIN)		R0 3580 (BIN)																				
D101 8701 (BIN)	-	8765 (BIN)	⇒	R1 -64 (BIN)																		
D102 3502 (BIN)				R2 -5263 (BIN)																		

6.2.12 \$ +, \$ + P

Процессор

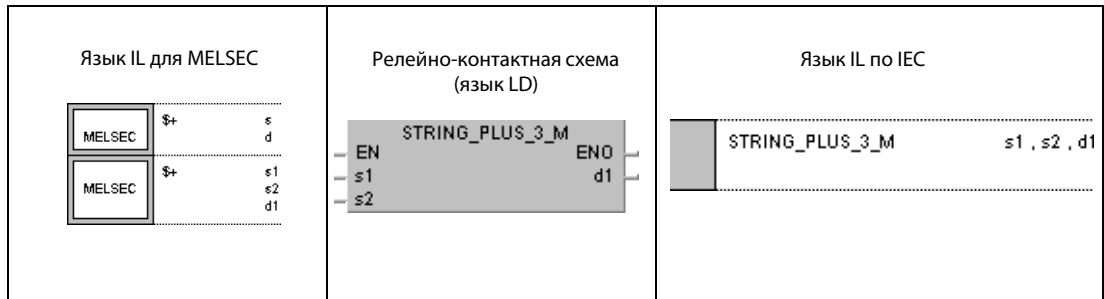
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

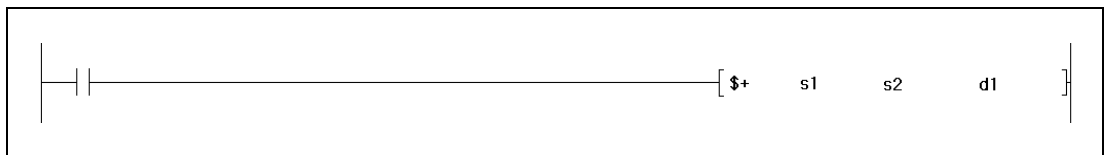
Операнды MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы \$	Иные U		
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	●	—	SM0	3
d	—	●	●	—	—	—	—	—	—		
s1	—	●	●	—	—	—	—	●	—	SM0	4
s2	—	●	●	—	—	—	—	●	—		
d1	—	●	●	—	—	—	—	—	—		

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
s	Данные или первый адрес, начиная с которого сохранены присоединяемые данные.	Строковая величина
d	Первый адрес, начиная с которого сохранены соединяемые данные и сохраняется результат соединения.	
s1	Данные или первый адрес, начиная с которого сохранены соединяемые данные.	
s2	Первый адрес, начиная с которого сохранены соединяемые данные.	
d1	Первый адрес, начиная с которого сохраняется результат соединения.	



**Принцип действия**

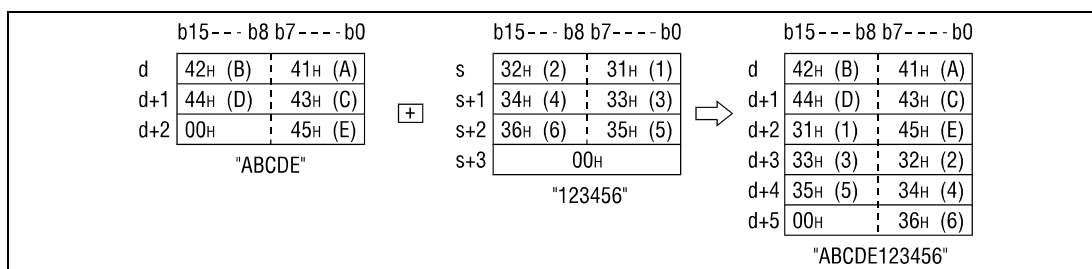
**Соединение строковых величин**

**§ + Команда соединения строковых величин**

- 1-й вариант:

Указанная в s строковая величина присоединяется к строковой величине, указанной в d. Объединенная строковая величина сохраняется в d.

Эта объединенная строковая величина начинается со знаков младших байтов строковой величины, которая перед этой операцией была указана в d, и заканчивается кодом "00н" строковой величины, указанной в s.

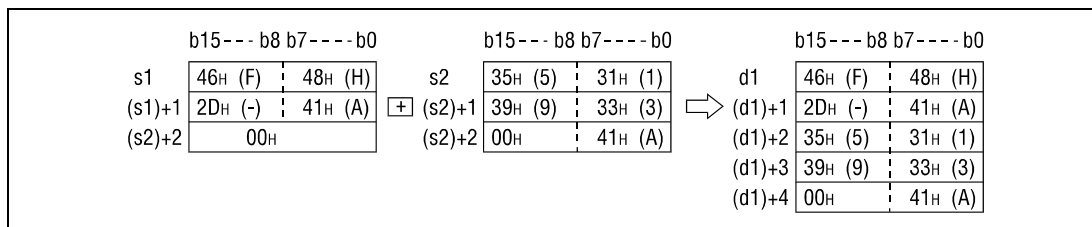


При соединении код "00н", обозначающий конец строковой величины в d, отбрасывается. Указанная в s строковая величина присоединяется вслед за последним знаком строковой величины, указанной в d.

- 2-й вариант:

Указанная в s1 строковая величина соединяется со строковой величиной, указанной в s2. Объединенная строковая величина сохраняется в d1.

Эта объединенная строковая величина начинается со знаков младших байтов строковой величины, хранящейся в s1, и заканчивается кодом "00н" строковой величины, хранящейся в s2.



При соединении код "00н", обозначающий конец строковой величины в s1, игнорируется. Указанная в s2 строковая величина присоединяется вслед за последним знаком строковой величины, указанной в s1.

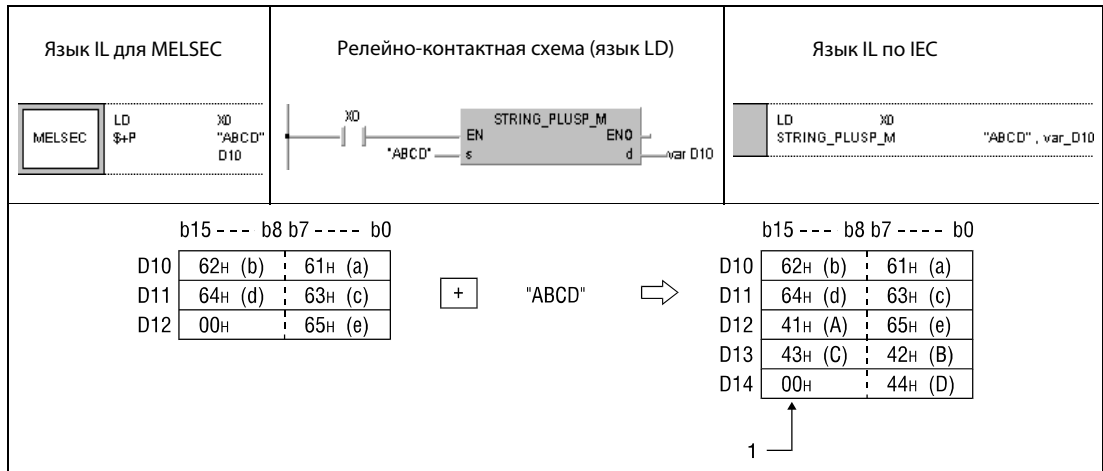
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Полная, объединенная строковая величина не может быть сохранена (код ошибки 4100).
- Предусмотренные для сохранения области данных s, s1 или s2 перекрываются с областями d или d1 (код ошибки 4101).

**Пример 1** S + P

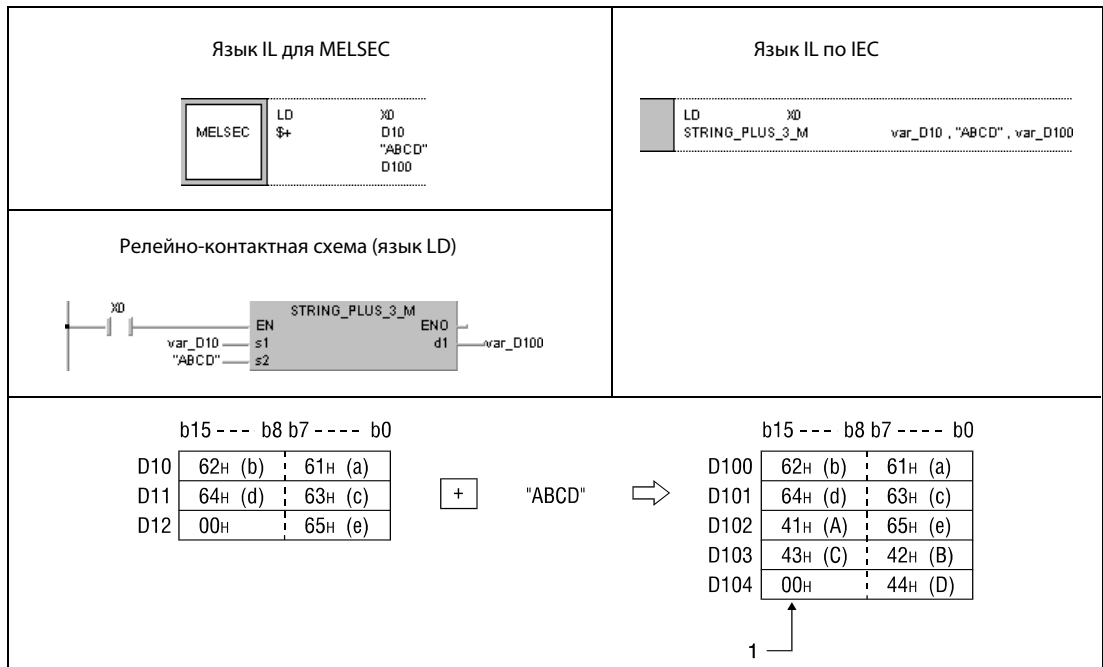
Следующая программа при положительном фронте сигнала X0 соединяет строковую величину, хранящуюся в регистрах D10...D12, со строковой величиной "ABCD". Объединенная строковая величина сохраняется в D10...D14.



<sup>1</sup> Для обозначения конца строковой величины в этот байт автоматически записывается код "00H".

**Пример 2** S +

Следующая программа при включенном входе X0 соединяет строковую величину, хранящуюся в регистрах D10...D12, со строковой величиной "ABCD". Объединенная строковая величина сохраняется в D101...D104.



<sup>1</sup> Для обозначения конца строковой величины в этот байт автоматически записывается код "00H".

**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию можно найти в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

### 6.2.13 INC, INCP, DEC, DECP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

	Операнды																Длина блока K1 ↓ K4	Шагов 3	Индекс 1	Флаг переноса M9012	Флаг ошибки M9011		
	битовые						словные (16 бит)						Константы		Указатели							Уровень N	
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V							K
d	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							●	●

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

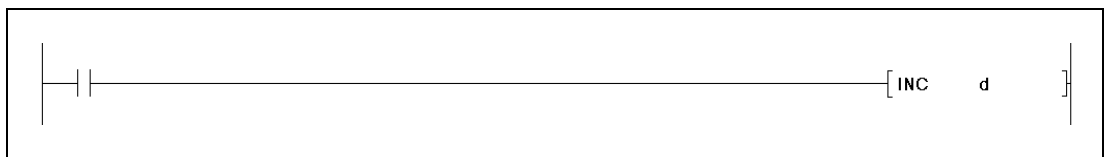
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□□		Спец. модули U□\G□□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
d	●	●	●	●	●	●	●	—	—	—	2

**GX IEC Developer**



**GX Developer**



**Переменные**

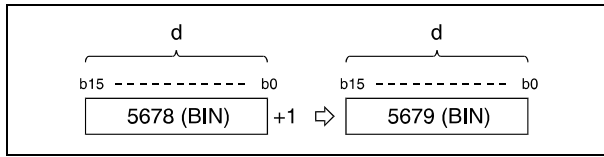
Операнд	Значение	Тип данных
d	Первый адрес, в отношении которого применяется команда INC или DEC.	BIN, 16 бит

**Принцип действия**

**Положительное и отрицательное приращение двоичных данных (16 бит)**

**INC Приращение двоичных данных (16 бит)**

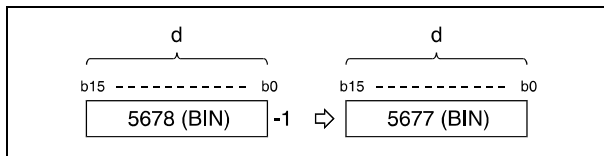
Указанный в d операнд (16 бит) повышается на 1.



Если при выполнении команды INC или INCP содержимое d равно 32767, в d записывается значение -32768.

**DEC Отрицательное приращение двоичных данных (16 бит)**

Указанный в d операнд (16 бит) понижается на 1.



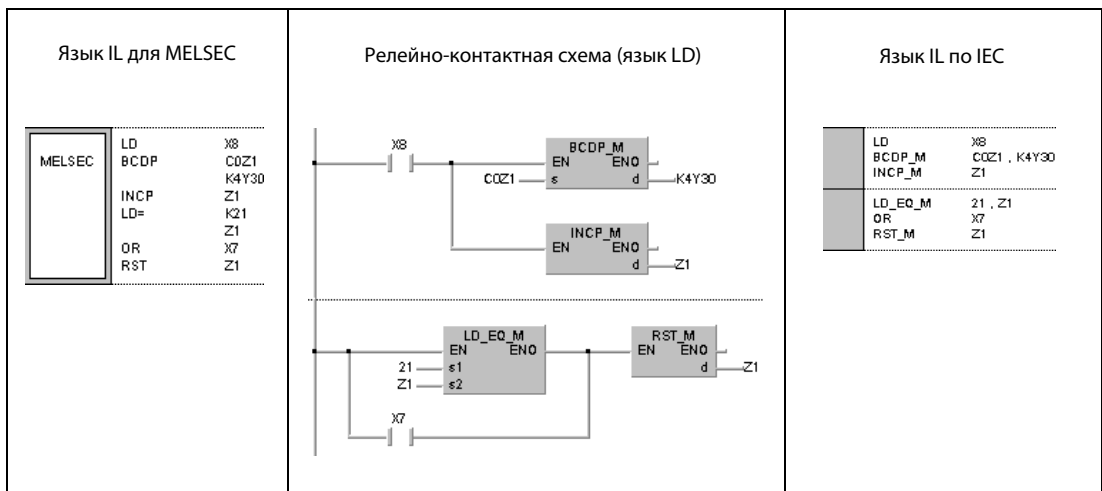
Если при выполнении команды DEC или DECP содержимое d равно 0, в d записывается -1.

Если при выполнении команды DEC или DECP содержимое d равно -32768, в d записывается значение 32767.

**Пример 1**

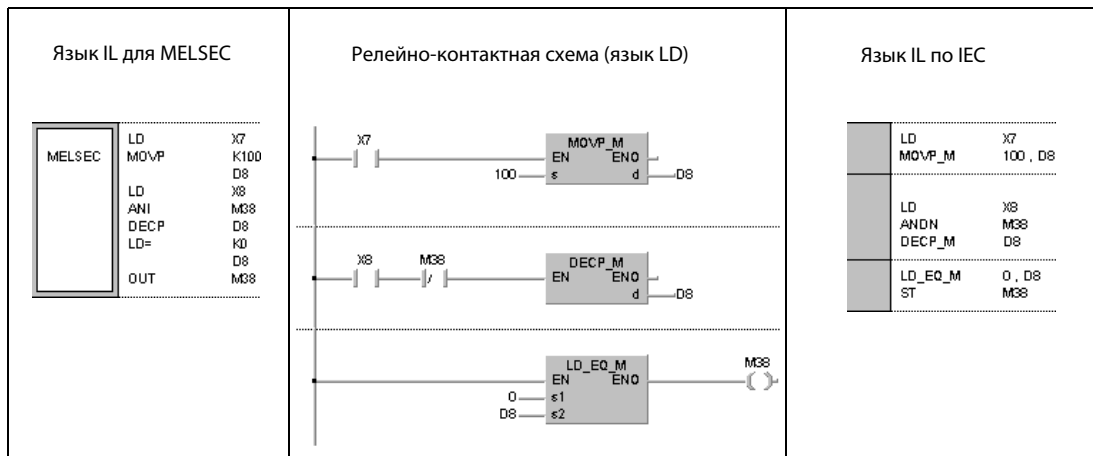
**INCP**

Следующая программа при каждом положительном фронте X8 выводит фактическое состояние счетчика C0...C20 (C0 плюс Z1) в двоично-десятичном виде на выходы Y30...Y3F (заданное значение счетчика равно 9999). Если Z1 равно 21 (LD = K21 Z1) или включается вход сброса X7, регистр Z1 устанавливается на 0 (RST Z1).



**Пример 2** DECP

Следующая программа содержит пример убывающего счетчика. При положительном фронте X7 в D8 записывается значение 100. При условии, что маркер M38 не установлен, при положительном фронте сигнала X8 значение в D8 понижается на 1. Маркер M38 устанавливается, как только D8 достигает 0.



### 6.2.14 DINC, DINCP, DDEC, DDEC P

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

	Операнды														Длина блока K1 ↓ K8	Шагов 3 <sup>2</sup>	Индекс ●	Флаг переноса M9012	Флаг ошибки M9011							
	битовые						словные (16 бит)						Константы							Указатели		Уровень N				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1						Z	V		K	H (16#)	P	I
d	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

<sup>1</sup> кроме процессоров AnN

<sup>2</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zп	Константы K, H (16#)			Иные U
	битовые	словные		битовые	словные						
d	●	●	●	●	●	●	●	—	—	—	2 <sup>1)</sup>

<sup>1</sup> Количество шагов программы зависит от типа центрального процессора и используемых операндов.

При использовании процессоров серии QnA или однопроцессорного модуля ЦП серии "Q": 2

При использовании многопроцессорного модуля ЦП серии "Q"

и внутренних словных операндов (кроме регистра файлов ZR): 3

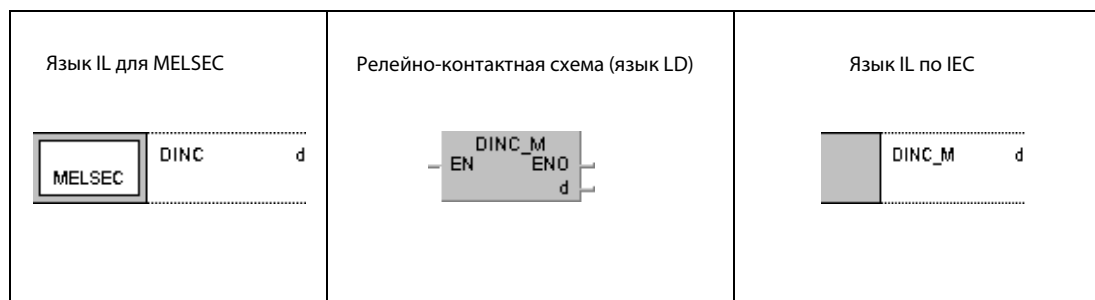
Константы: 3

Битовые операнды с адресом, кратным 16, имеющие обозначение битового блока K8

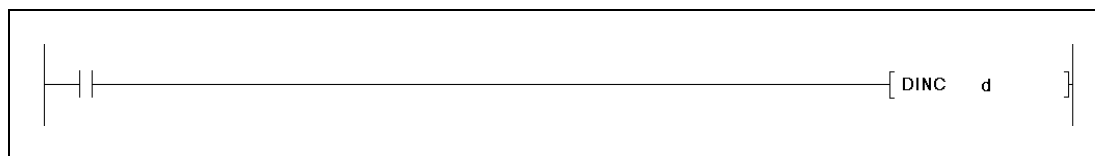
и не индексруемые: 3

Прочие операнды кроме названных: 2

GX IEC Developer



GX Developer



Переменные

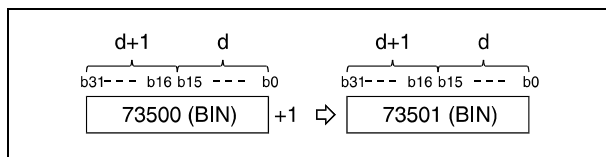
Операнд	Значение	Тип данных
d	Первый адрес, в отношении которого применяется команда DINC или DDEC.	BIN, 32 бита

**Принцип действия**

**Положительное и отрицательное приращение двоичных данных (32 бита)**

**DINC Приращение двоичных данных (32 бита)**

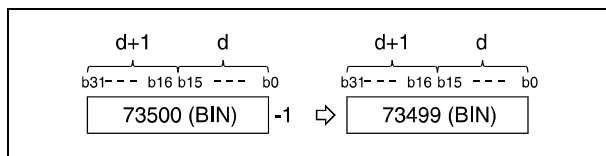
Указанный в d операнд (32 бита) повышается на 1.



Если при выполнении команды DINC или DINCP содержимое d равно 2147483647, в d записывается значение -2147483648.

**DDEC Отрицательное приращение двоичных данных (32 бита)**

Указанный в d операнд (32 бита) понижается на 1.



Если при выполнении команды DDEC или DDECP содержимое d равно 0, в d записывается значение -1.

Если при выполнении команды DDEC или DDECP содержимое d равно -2147483648, в d записывается значение 2147483647.

**Пример 1**

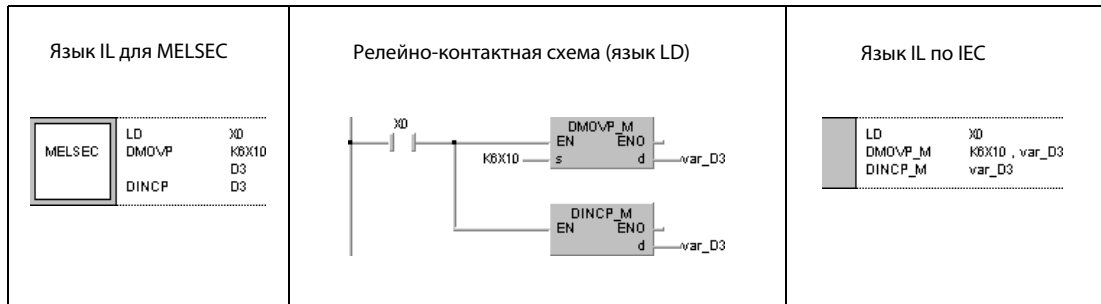
**DINCP**

Следующая программа при положительном фронте сигнала X0 повышает значение в D0 на 1.



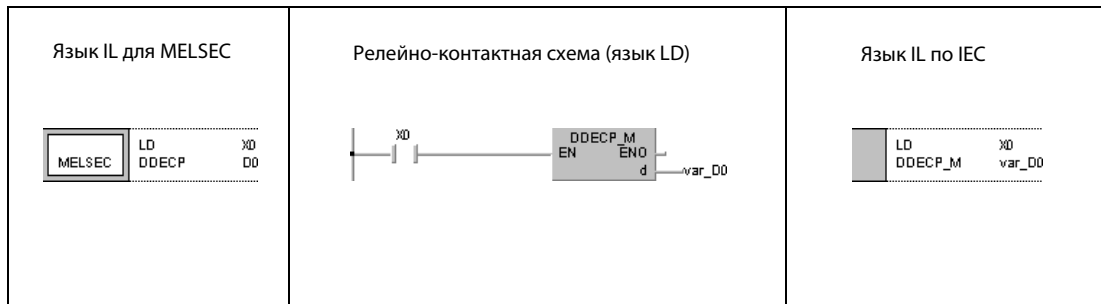
**Пример 2** DINC

Следующая программа при положительном фронте сигнала X0 повышает значения данных с X10 по X27 на 1 и сохраняет результаты в регистрах D3 и D4.



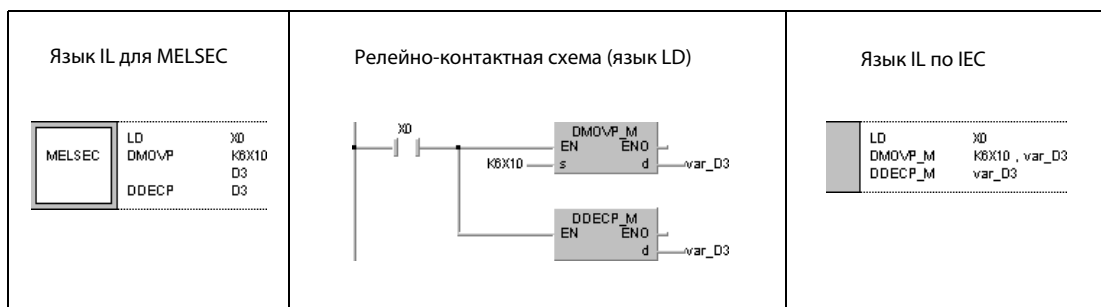
**Пример 3** DDEC

Следующая программа при положительном фронте сигнала X0 понижает значение в D0 на 1.



**Пример 4** DDECP

Следующая программа при положительном фронте сигнала X0 понижает значения данных с X10 по X27 на 1 и сохраняет результат в D3 и D4.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.



## 6.3 Команды конвертирования

Описываемые в этом разделе команды преобразуют различные форматы данных.

**ПРИМЕЧАНИЕ** В редакторах IEC следует использовать команды по стандарту IEC.

Конвертирование	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC
BIN (16/32 бита) ↓ BCD (4/8 знаков)	BCD	BCD_M
	BCDP	BCDP_M
	DBCD	DBCD_M
	DBCDP	DBCDP_M
BCD (4/8 знаков) ↓ BIN (16/32 бита)	BIN	BIN_M
	BINP	BINP_M
	DBIN	DBIN_M
	DBINP	DBINP_M
BIN (16/32 бита) ↓ число с плавающей запятой	FLT	FLT_M
	FLTP	FLTP_M
	DFLT	DFLT_M
	DFLTP	DFLTP_M
Число с плавающей запятой ↓ BIN (16/32 бита)	INT	INT_MD
		INT_E_MD
	INTP	INT_P_MD
		INT_P_E_MD
	DINT	DINT_MD
		DINT_E_MD
DINTP	DINT_P_MD	
	DINT_P_E_MD	
BIN, 16 бит ↓ BIN, 32 бита	DBL	DBL_M
	DBLP	DBLP_M
BIN, 32 бита ↓ BIN, 16 бит	WORD	WORD_M
	WORDP	WORDP_M
BIN (16/32 бита) ↓ Код Грея	GRY	GRY_M
	GRYP	GRYP_M
	DGRY	DGRY_M
	DGRYP	DGRYP_M
Код Грея ↓ BIN (16/32 бит)	GBIN	GBIN_M
	GBINP	GBINP_M
	DGBIN	DGBIN_M
	DGBINP	DGBINP_M
Изменение арифметического знака BIN (16/32 бит) (преобразование в дополнительный код)	NEG	NEG_M
	NEGP	NEGP_M
	DNEG	DNEG_M
	DNEGP	DNEGP_M
Изменение арифметического знака числа с плавающей запятой	ENEG	ENEG_M
	ENEGP	ENEGP_M

Конвертирование	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC
Блоки BIN (16 бит) ↓ блоки BCD (4 знака)	BKBCD	BKBCD_M
	BKBCDP	BKBCDP_M
Блоки BCD (4 знака) ↓ блоки BIN (16 бит)	BKBIN	BKBIN_M
	BKBINP	BKBINP_M

### 6.3.1 BCD, BCDP, DBCD, DBCDP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

Операнды															Длина блока	Шагов	Индекс	Флаг																					
битовые					словные (16 бит)					Константы		Указатели		Уровень				M9012	M9010																				
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V	K	H (16#)	P	I	N														
BCD																																							
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●											K1 ↓ K4	5	1	●	●									
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●																								
DBCD																																							
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●											K1 ↓ K8	9	1	●	●									
d		●	●	●	●	●	●	●	●	●	●	●	●	●																									

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

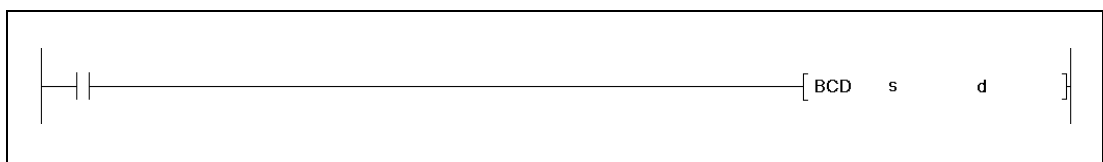
**Операнды MELSEC Q**

Операнды									Флаг ошибки	Шагов
Внутренние операнды (систем., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	—	SM0	3
d	●	●	●	●	●	●	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Двоичные данные или операнд, в котором сохранены двоичные данные.	BIN, 16/32 бита
d	Операнд, в котором сохраняются двоично-десятичные данные.	BCD, 4/8 знаков

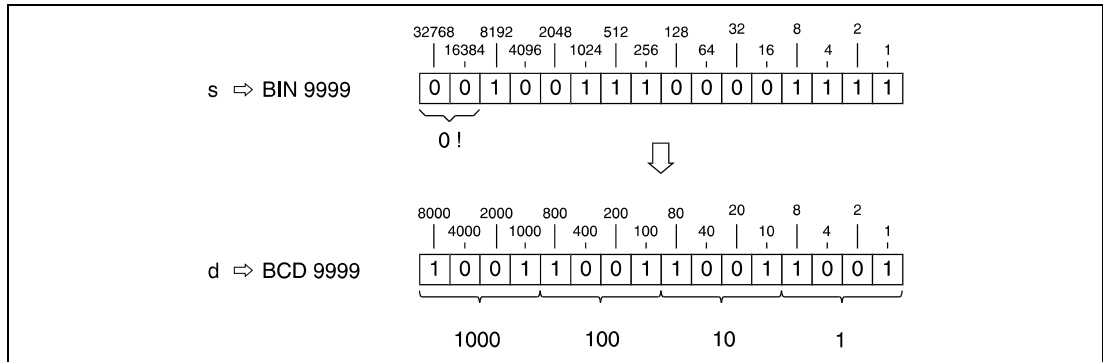
**Принцип действия**

**Конвертирование данных из двоичного формата в двоично-десятичный (4/8 знаков)**

**BCD Конвертирование двоичных данных в двоично-десятичные (4-значные)**

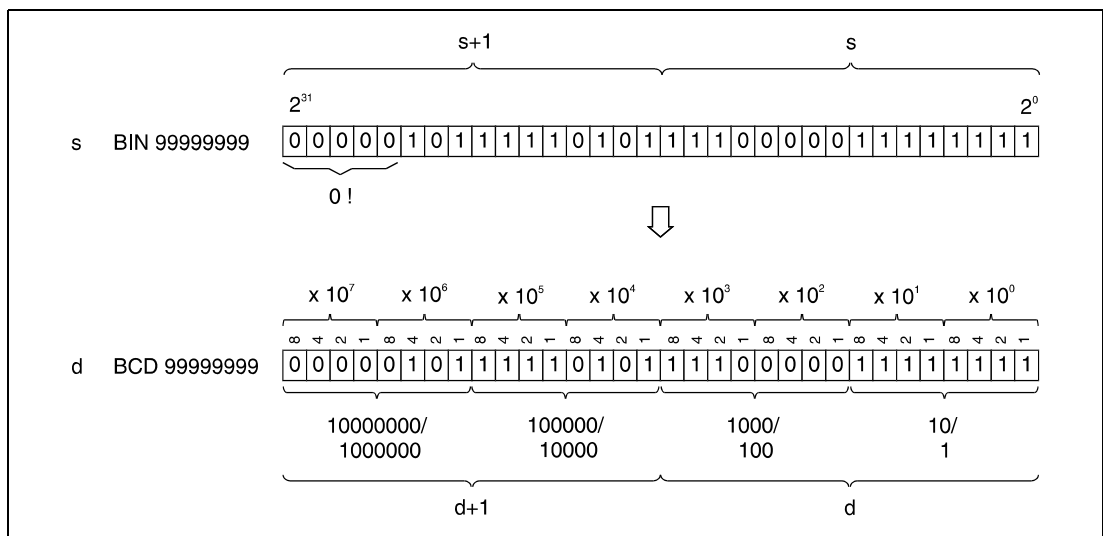
Команда BCD преобразует указанное в s двоичное значение данных (от 0 до 9999) в двоично-десятичное значение и передает результат в операнд, указанный в d.

В случае 4-значного значения данных два старших бита двоичного значения в s должны иметь состояние 0.



**DBCD Конвертирование двоичных данных в двоично-десятичные (8-значные)**

Команда DBCD работает аналогично команде BCD. Однако двоичное значение данных может быть 8-значным (от 0 до 99999999). Пять старших битов двоичного значения в s должны иметь состояние 0.



**Источники ошибок**

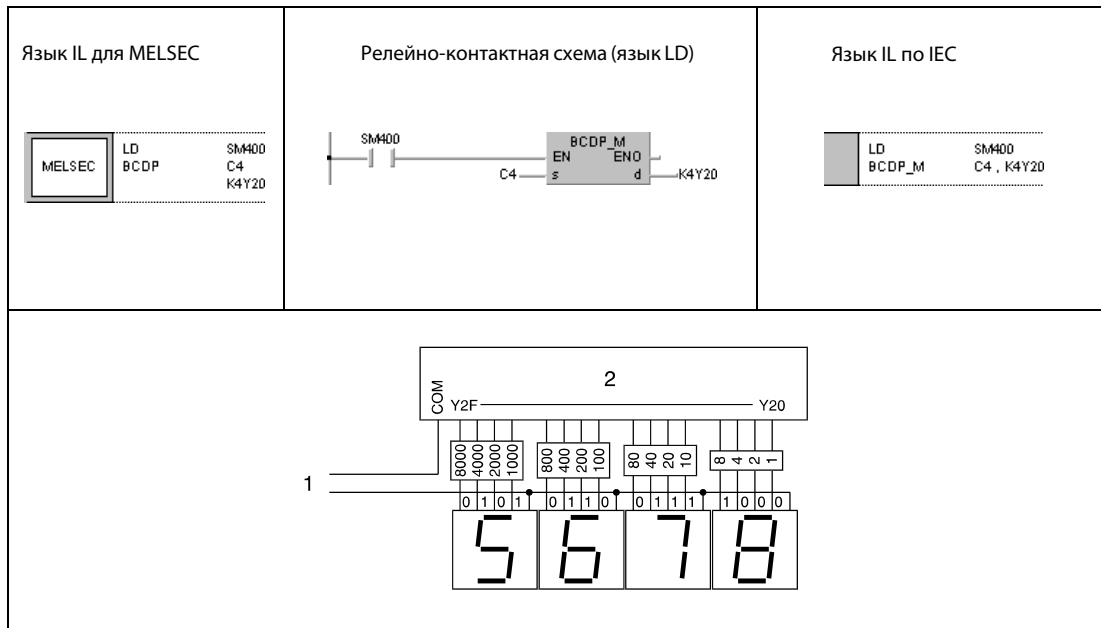
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- При применении команды BCD исходные данные в s находятся вне диапазона от 0 до 9999 (серии "Q"/"System Q" = код ошибки 4100).
- При применении команды DBCD исходные данные в s + 1 или s находятся вне диапазона от 0 до 99999999 (серии "Q"/"System Q" = код ошибки 4100).

**Пример**

BCDP

Следующая программа при положительном фронте сигнала SM400 выводит фактическое значение C4 (5678) на выходы Y20...Y2F для двоично-десятичного дисплея.



<sup>1</sup> электропитание

<sup>2</sup> выходной модуль

### 6.3.2 BIN, BINP, DBIN, DBINP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

Операнды															Длина блока	Шагов	Индекс	Флаг переноса		Флаг ошибки											
битовые					словные (16 бит)					Константы		Указатели		Уровень				M9012	M9010												
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z				V	K	H (16#)	P	I	N								
BIN																															
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●												K1 ↓ K4	5	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●														●		
DBIN																															
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●												K1 ↓ K8	9	●		●
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●														●		

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

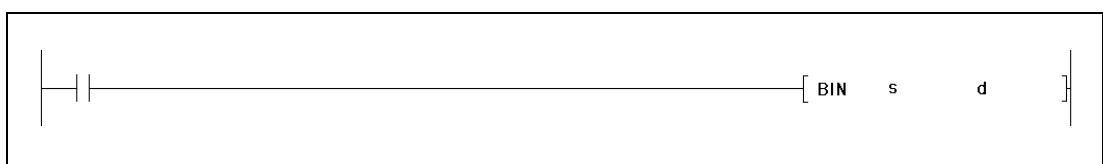
Операнды MELSEC Q

Операнды									Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непос. адр. J□□□		Спец. модули U□NG□□	Индексные регистры Zn	Константы K, H (16#)	Иные U		
битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	—	SM0	3
d	●	●	●	●	●	●	—	—		

GX IEC Developer



GX Developer



Переменные

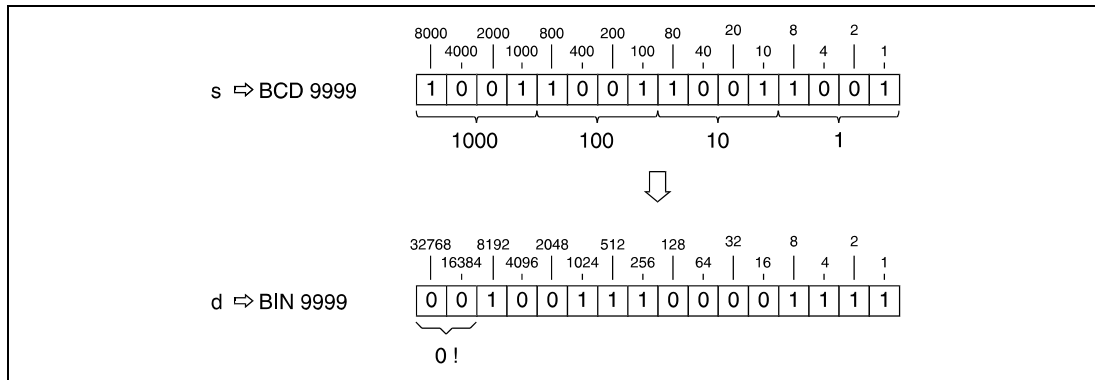
Операнд	Значение	Тип данных
s	Двоично-десятичные данные или операнд, в котором сохранены двоично-десятичные данные.	BCD, 4/8 знаков
d	Операнд, в котором сохраняются двоичные данные.	BIN, 16/32 бита

**Принцип действия**

**Конвертирование двоично-десятичных данных (4/8 знаков) в двоичные**

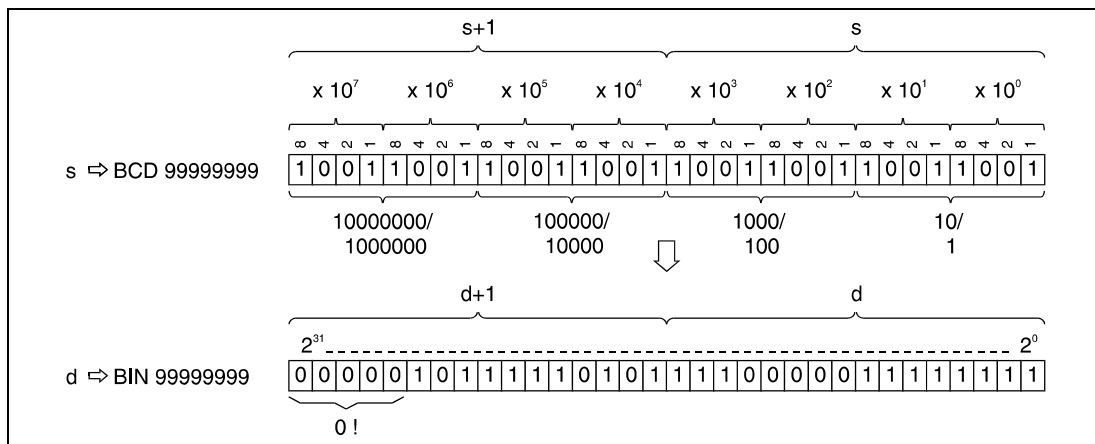
**BIN Конвертирование двоично-десятичных данных (4 знака) в двоичные**

Команда BIN преобразует указанное в s двоично-десятичное значение данных (от 0 до 9999) в двоичное значение и передает результат в операнд, указанный в d. В случае 4-значного десятичного числа два старших бита двоичного значения в d должны быть всегда равны 0.



**DBIN Конвертирование двоично-десятичных данных (8 знаков) в двоичные**

Команда DBIN действует аналогично команде BIN. Однако двоично-десятичное значение данных может быть 8-значным (от 0 до 99999999). Пять старших битов двоичного значения в d должны иметь состояние 0.



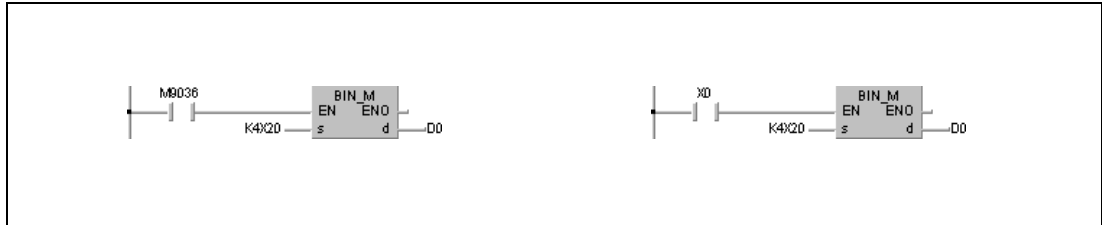
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Один из знаков исходного значения данных в s не находится в диапазоне между 0 и 9.

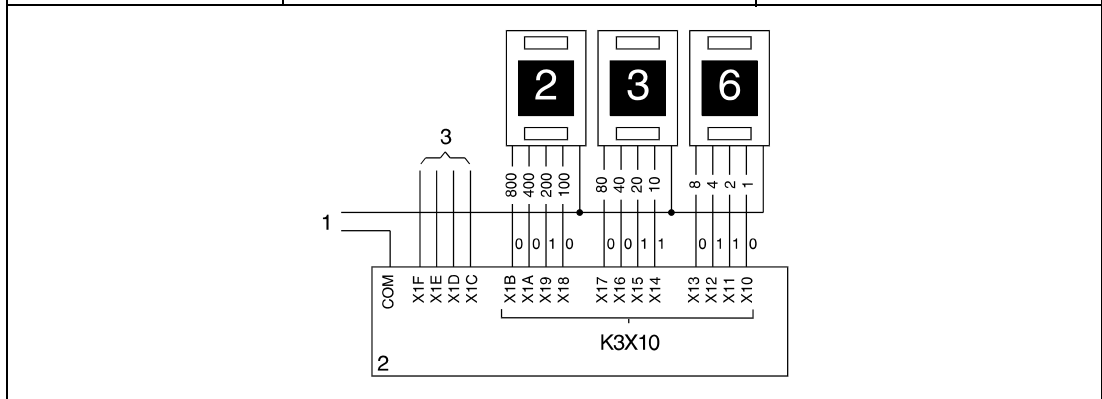
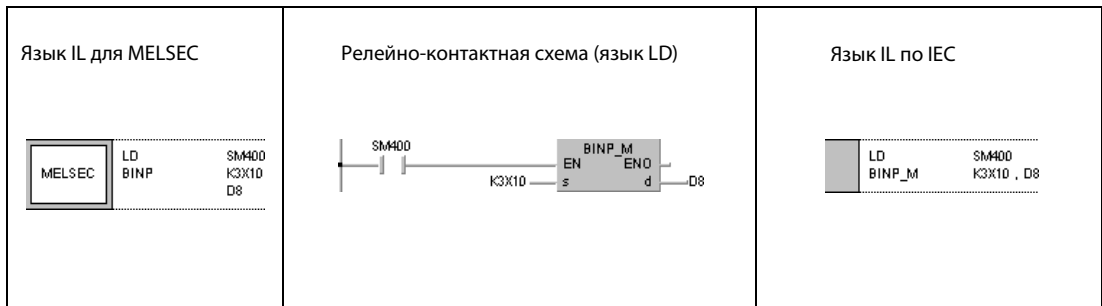
При использовании центрального процессора Q сообщение об ошибке можно подавить, установив SM722. Однако если превышен допустимый диапазон для s, команда не выполняется независимо от состояния SM722.

**ПРИМЕЧАНИЕ** Если для команды BIN или DBIN в качестве входного условия используется специальный маркер M9036 или M9037, то при некоторых обстоятельствах в связи с задержкой переключения двоично-десятичной индикации может возникнуть ошибка выполнения программы. В этом случае двоичное конвертирование рекомендуется выполнять лишь после установки двоично-десятичных данных с помощью обычного входного контакта (это относится только к серии "А").



**Пример 1** BINP

Следующая программа преобразует двоично-десятичные данные входов X10...X1B в двоичный формат и при положительном фронте сигнала SM400 сохраняет результат в D8.

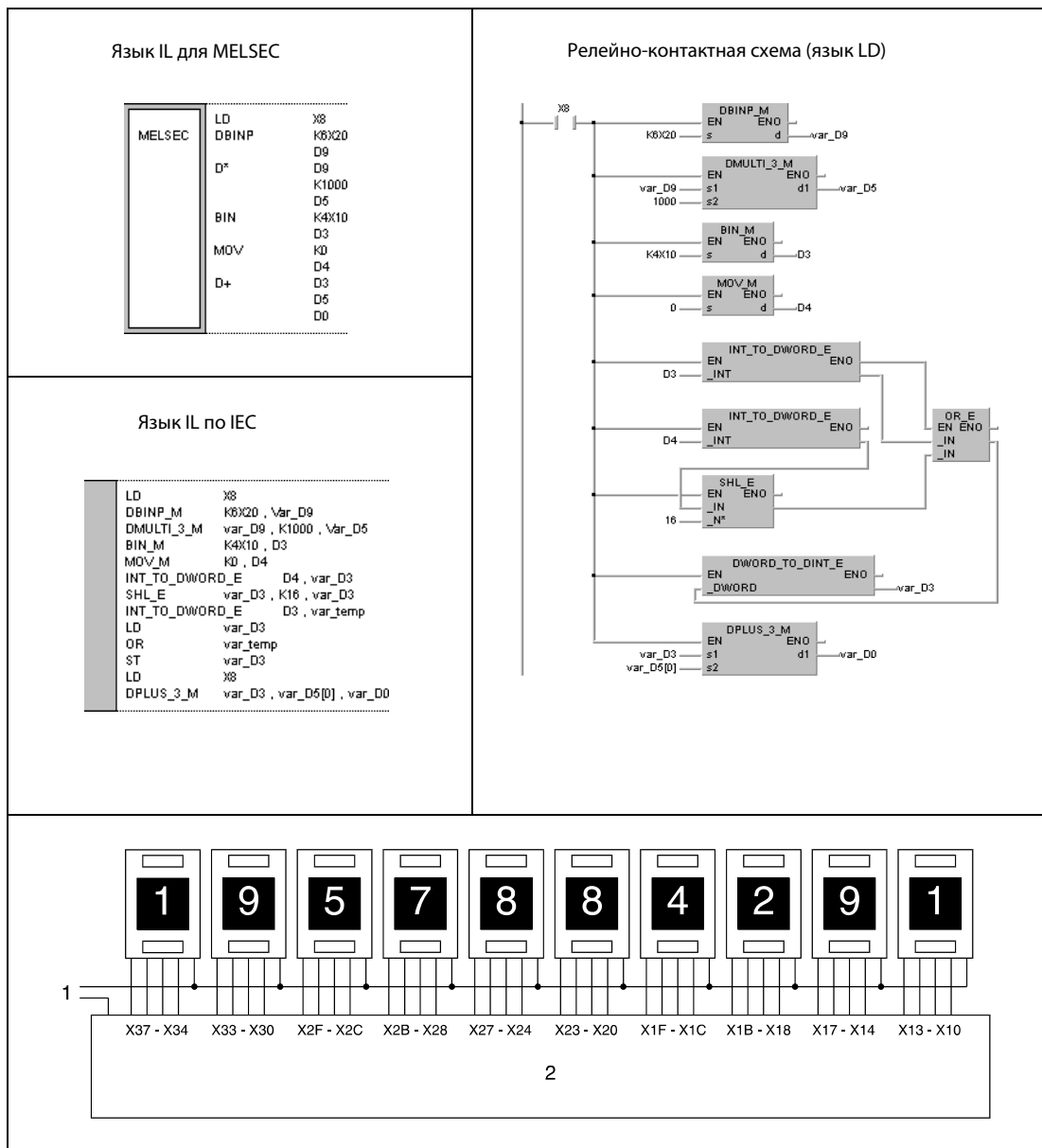


- 1 электропитание
- 2 входной модуль
- 3 входы для свободного использования



**Пример 2** DBINP

Следующая программа при положительном фронте сигнала X8 преобразует двоично-десятичные данные входов X10...X37 в двоичный формат. Результат сохраняется в D0 и D1.



- 1 электропитание
- 2 входной модуль

**ПРИМЕЧАНИЯ**

Если на входах X10...X37 находятся двоично-десятичные значения выше 2147483647, то эти значения находятся вне диапазона, который может вместить 32-битный операнд! В этом случае значения в D0 и D1 становятся отрицательными. Дополнительную информацию на эту тему можно найти в разд. 3.4 „Программирование переменных“ этого руководства.

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию на эту тему см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

### 6.3.3 FLT, FLTP, DFLT, DFLTP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

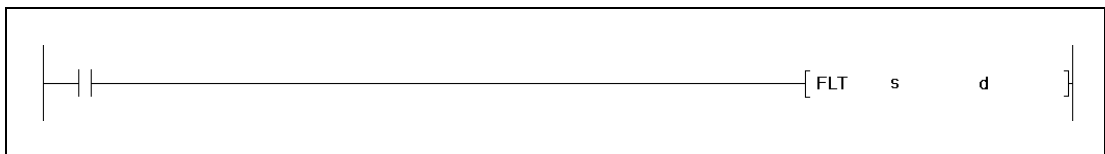
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	—	—	3	
d	—	●	●	—	●	—	—	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

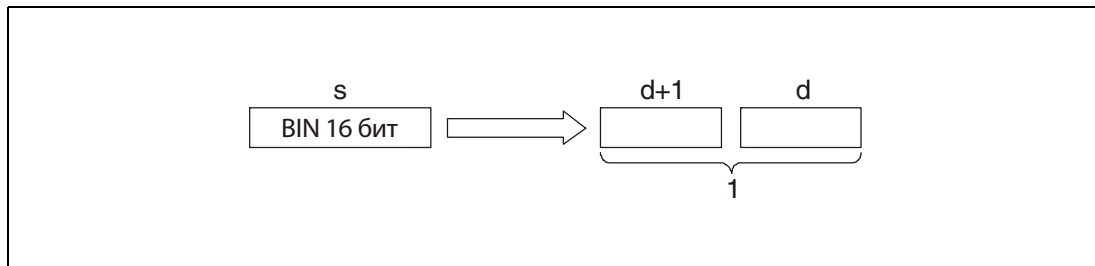
Операнд	Значение	Тип данных
s	Операнд, в котором сохранены конвертируемые данные типа BIN.	BIN, 16/32 бита
d	Операнд, в котором сохраняется конвертированное число с плавающей запятой.	REAL

**Принцип действия**

**Преобразование 16/32-битных двоичных чисел в числа с плавающей запятой**

**FLT Преобразование двоичных данных (16 бит) в числа с плавающей запятой**

Указанные в s двоичные данные преобразуются в число с плавающей запятой. Результат сохраняется в d.

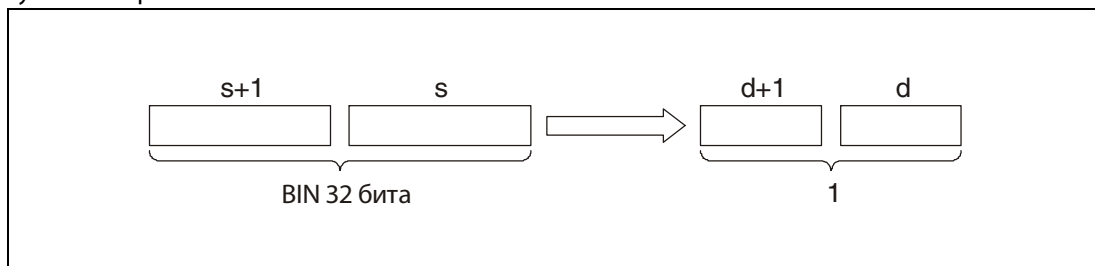


<sup>1</sup> число с плавающей запятой (данные типа REAL)

Указанное в s значение данных должно находиться в диапазоне между -32768 и 32767.

**DFLT Конвертирование двоичных данных (32 бита) в числа с плавающей запятой**

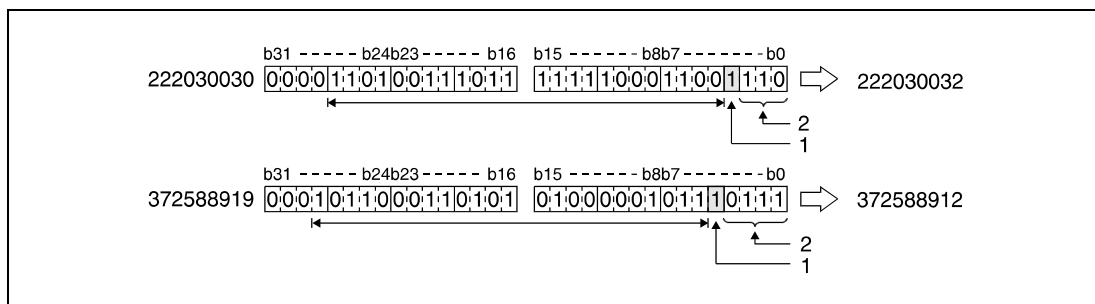
32-битные двоичные данные, указанные в s, преобразуются в число с плавающей запятой. Результат сохраняется в d.



<sup>1</sup> Число с плавающей запятой (данные типа REAL)

Указанное в s и s + 1 значение должно находиться в диапазоне между -2147483648 и 2147483647.

В связи с тем, что числа с плавающей запятой обрабатываются в 32-битном режиме, при двоичном отображении ширина данных уменьшается до 24 битов, а при десятичном отображении – до 7 знаков (приблизительно). Перед конвертацией двоичное значение округляется до ширины 25 битов. Все биты после 25-го отбрасываются. Если округленное значение (целое) находится вне диапазона, который можно отобразить 24 битами (от -16777216 до 16777215), во время преобразования может возникнуть ошибка.

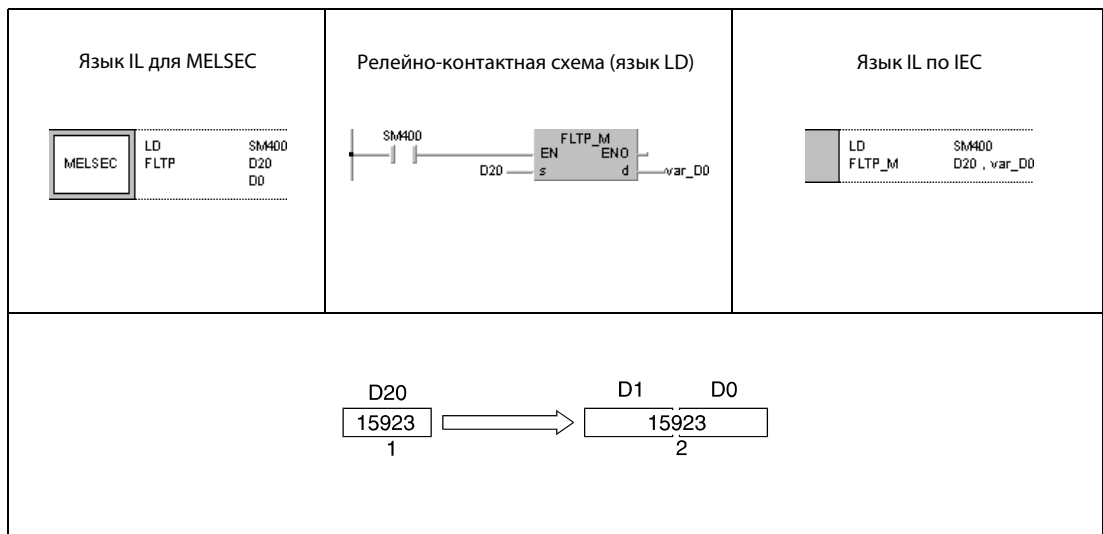


<sup>1</sup> округленный разряд

<sup>2</sup> отброшенные биты

**Пример 1** FLTP

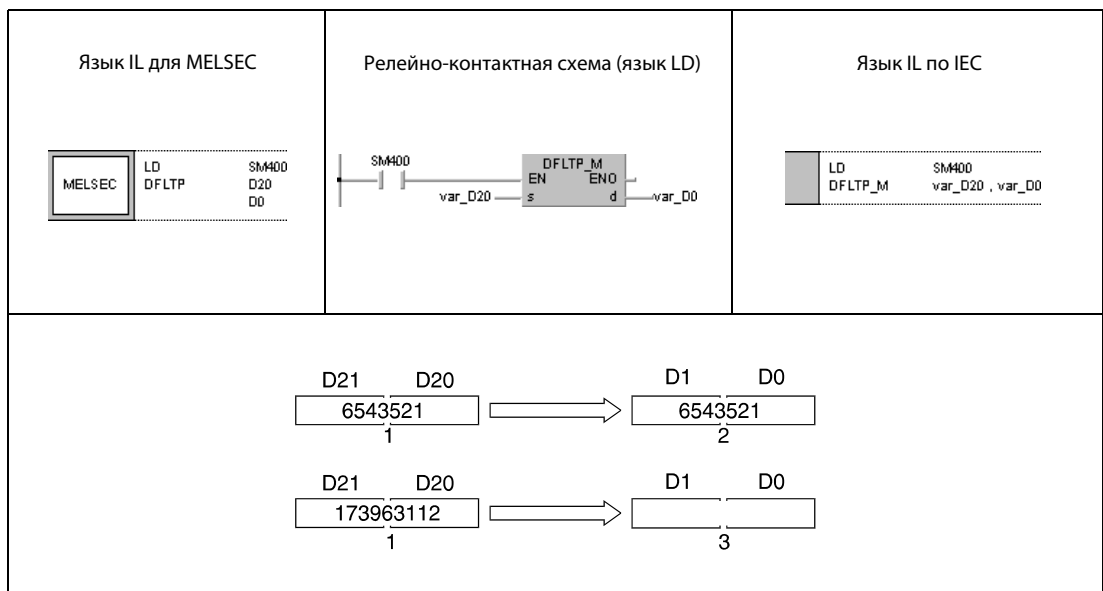
Следующая программа при положительном фронте сигнала SM400 преобразует хранящееся в D20 16-битное двоичное число в число с плавающей запятой и сохраняет результат в D0 и D1.



- <sup>1</sup> двоичные данные
- <sup>2</sup> число с плавающей запятой (данные типа REAL)

**Пример 2** DFLTP

Следующая программа при положительном фронте сигнала SM400 преобразует хранящееся в D20 и D21 32-битное двоичное число в число с плавающей запятой и сохраняет результат в D0 и D1.



- <sup>1</sup> двоичные данные
- <sup>2</sup> число с плавающей запятой (данные типа REAL)
- <sup>3</sup> Ошибка преобразования, так как имеются 7 значащих разрядов.

**ПРИМЕЧАНИЕ** *Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.*

### 6.3.4 INT, INTP, DINT, DINTP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

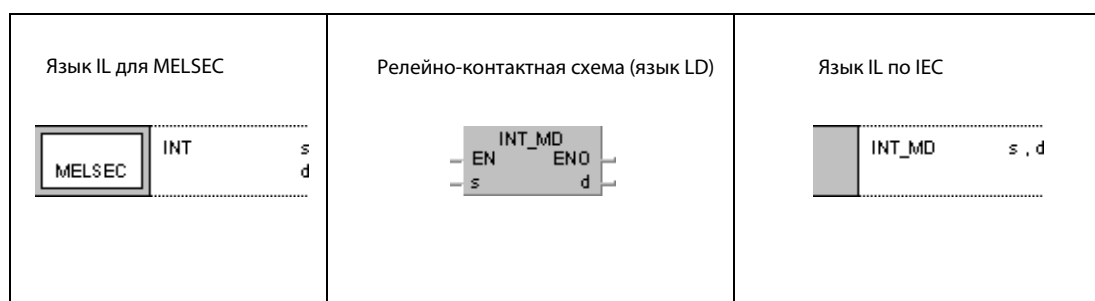
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

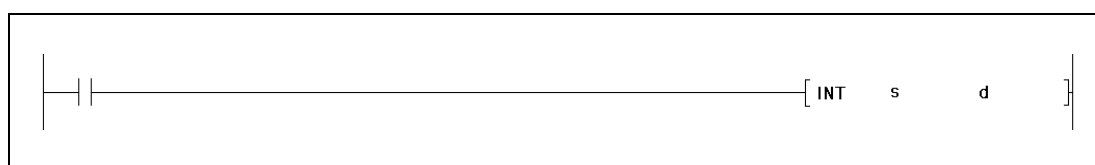
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы E			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	●	●	—	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

**GX IEC Developer**



**GX Developer**

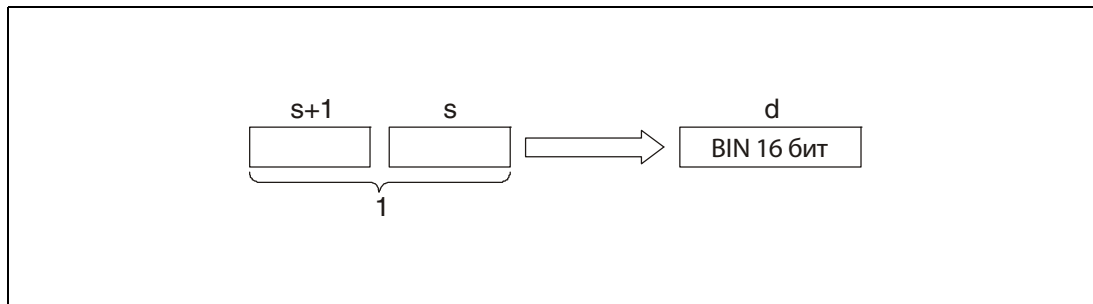


**Переменные**

Операнд	Значение	Тип данных
s	Операнд, в котором сохранено конвертируемое число с плавающей запятой.	REAL
d	Операнд, в котором сохраняются конвертированные данные типа BIN.	BIN, 16/32 бита

**Принцип действия****Преобразование чисел с плавающей запятой в 16/32-битные двоичные данные****INT Преобразование чисел с плавающей запятой в 16-битные двоичные данные**

Число с плавающей запятой, хранящееся в  $s$ , преобразуется в 16-битное двоичное число и результат сохраняется в  $d$ .



<sup>1</sup> число с плавающей запятой (данные типа REAL)

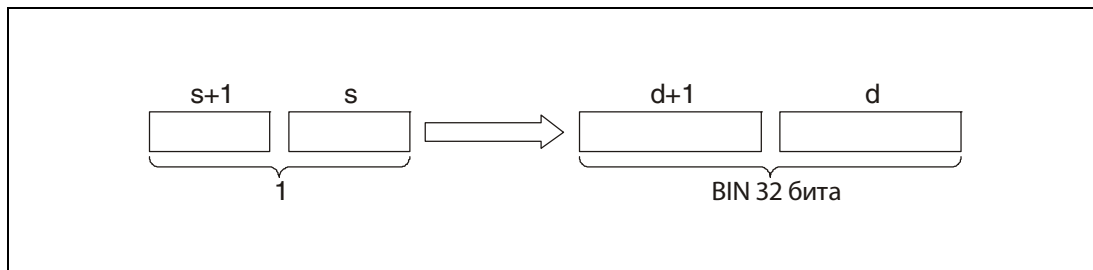
Введенное в  $s$  и  $s + 1$  число с плавающей запятой должно быть в диапазоне между  $-32768$  и  $32767$ .

Полученное целое число сохраняется в  $d$  в 16-битном двоичном формате.

Сохраненное в  $d$  значение представляет собой число с плавающей запятой, округленное до разряда перед запятой и преобразованное.

**DINT Преобразование чисел с плавающей запятой в 32-битные двоичные данные**

Число с плавающей запятой, хранящееся в  $s$ , преобразуется в 32-битное двоичное число и результат сохраняется в  $d$ .



<sup>1</sup> число с плавающей запятой (данные типа REAL)

Хранящееся в  $s$  и  $s + 1$  число с плавающей запятой должно быть в диапазоне между  $-2147483648$  и  $2147483647$ .

Полученное целое число сохраняется в 32-битном двоичном формате.

Сохраненное в  $d$  значение представляет собой число с плавающей запятой, округленное до разряда перед запятой и преобразованное.

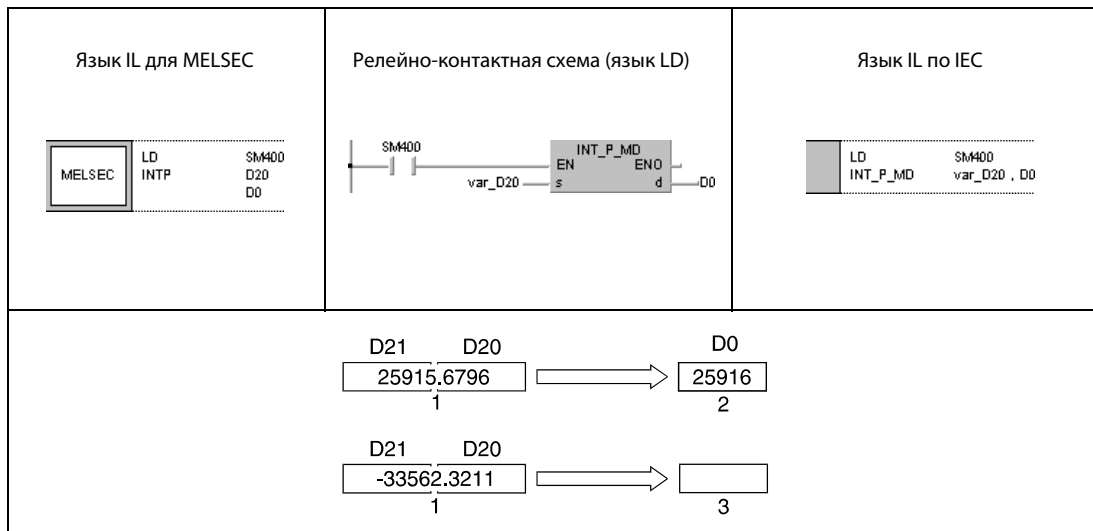
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- При применении команды INT указанное в  $s$  значение находится вне диапазона от  $-32768$  до  $32767$ .
- При применении команды DINT указанное в  $s$  значение находится вне диапазона от  $-2147483648$  до  $2147483647$ .

**Пример 1** INTP

Следующая программа при положительном фронте сигнала SM400 преобразует хранящееся в D20 и D21 число с плавающей запятой в 16-битное двоичное число и сохраняет результат по адресу D0.



<sup>1</sup> число с плавающей запятой (данные типа REAL)

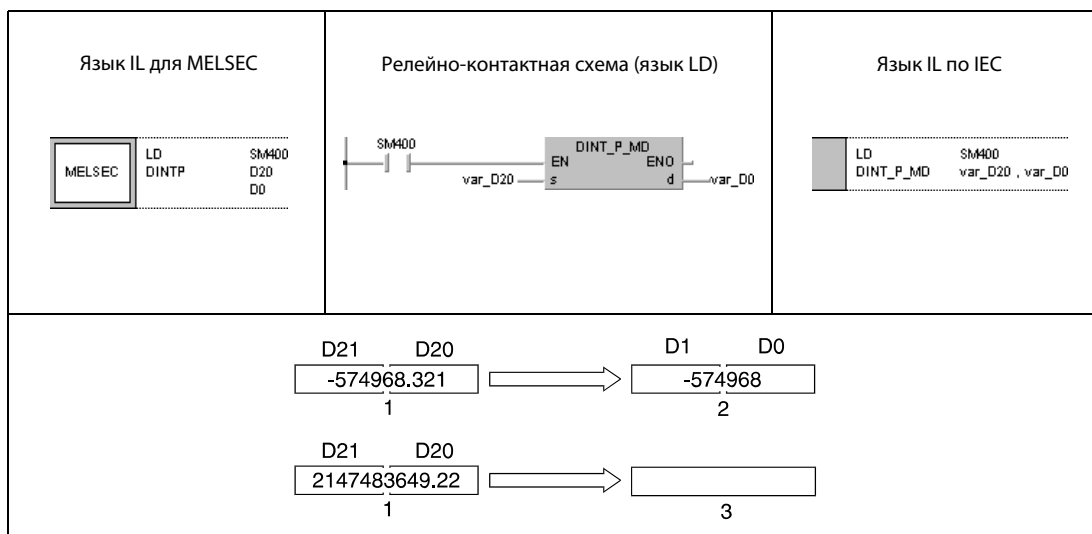
<sup>2</sup> двоичные данные

<sup>3</sup> Конвертируемое значение находится вне действительного диапазона для команды INT. В результате этого преобразование не происходит и сигнализируется ошибка.



**Пример 2** DINTP

Следующая программа при положительном фронте сигнала SM400 преобразует хранящееся в D20 и D21 число с плавающей запятой в 32-битное двоичное число. Результат сохраняется в D0 и D1.



<sup>1</sup> число с плавающей запятой (данные типа REAL)

<sup>2</sup> двоичные данные

<sup>3</sup> Конвертируемое значение находится вне действительного диапазона для команды DINT. В результате этого преобразование не происходит и сигнализируется ошибка.

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

### 6.3.5 DBL, DBLP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

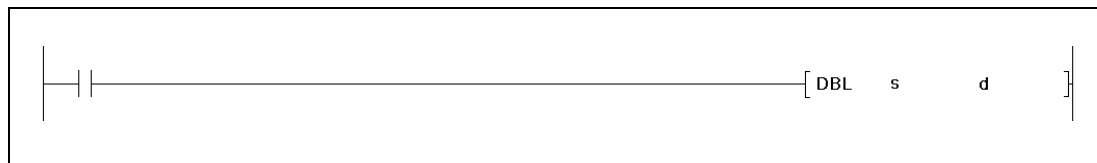
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	●	—	—	3
d	●	●	●	●	●	●	—	—	—	—	

**GX IEC Developer**



**GX Developer**



**Переменные**

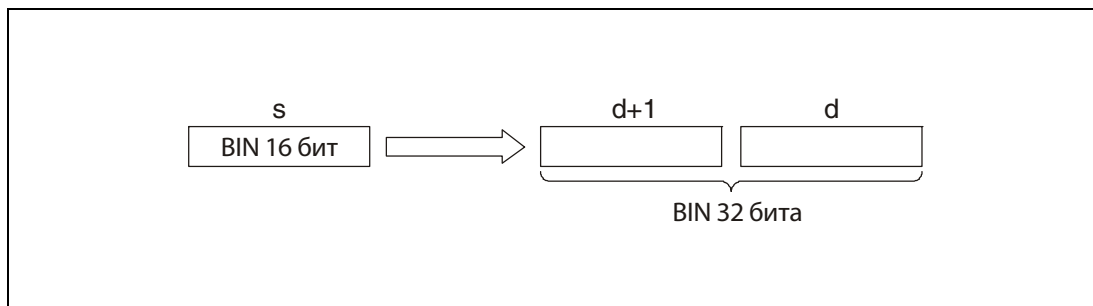
Операнд	Значение	Тип данных
s	Операнд, в котором сохранены конвертируемые данные типа BIN.	BIN, 16 бит
d	Операнд, в котором сохраняются конвертированные данные типа BIN.	BIN, 32 бита

**Принцип действия**

**Преобразование 16-битных двоичных данных в 32-битные двоичные данные**

**DBL Преобразование 16-битных двоичных данных в 32-битные двоичные данные**

16-битные двоичные данные, хранящиеся в s, преобразуются в 32-битные двоичные данные. Результат с арифметическим знаком сохраняется в d.



**Пример**

**DBLP**

Следующая программа при положительном фронте сигнала X20 преобразует хранящееся в D100 16-битное двоичное число в 32-битное двоичное число. Результат сохраняется в R0 и R1.

<p>Язык IL для MELSEC</p> <pre style="font-family: monospace; border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> MELSEC  LD      X20          DBLP   D100          R0</pre>	<p>Релейно-контактная схема (язык LD)</p>	<p>Язык IL по IEC</p> <pre style="font-family: monospace; border: 1px solid black; padding: 5px; width: fit-content; margin: auto;"> LD      X20 DBLP_M  D100 , var_R0</pre>

**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

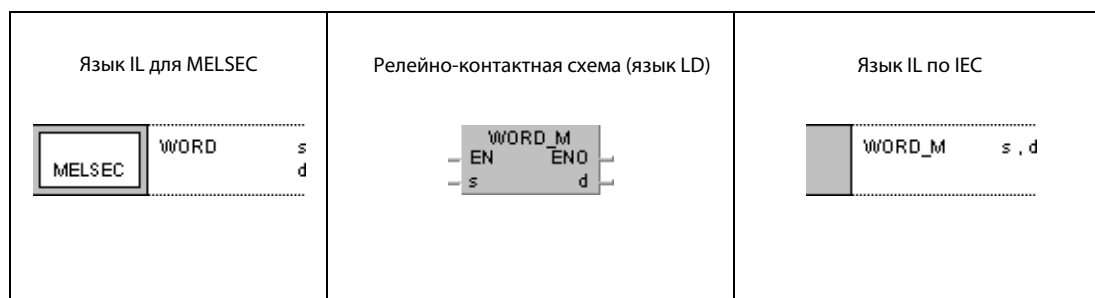
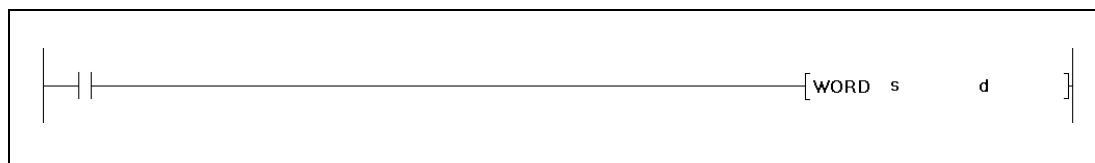
## 6.3.6 WORD, WORDP

## Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Операнды  
MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	●	—	SM0	3
d	●	●	●	●	●	●	—	—	—		

GX IEC  
DeveloperGX  
Developer

## Переменные

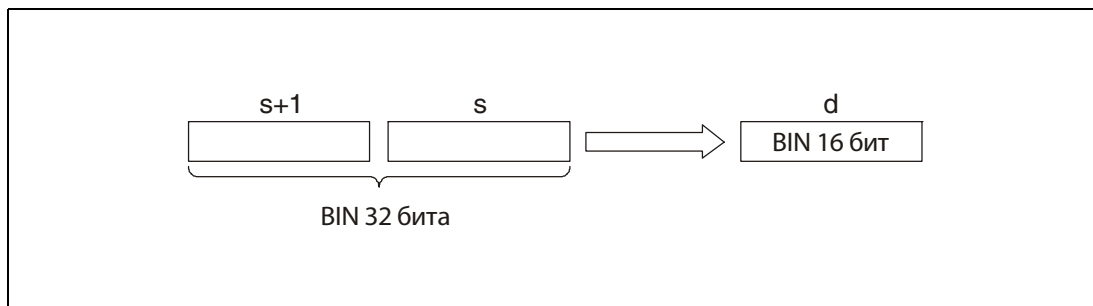
Операнд	Значение	Тип данных
s	Операнд, в котором сохранены конвертируемые двоичные данные.	BIN, 32 бита
d	Операнд, в котором сохраняются конвертированные двоичные данные.	BIN, 16 бит

**Принцип действия**

**Преобразование 32-битных двоичных данных в 16-битные двоичные данные**

**WORD Преобразование 32-битных двоичных данных в 16-битные двоичные данные**

32-битные двоичные данные, хранящиеся в s, преобразуются в 16-битные двоичные данные. Результат с арифметическим знаком сохраняется в d.



**Источники ошибок**

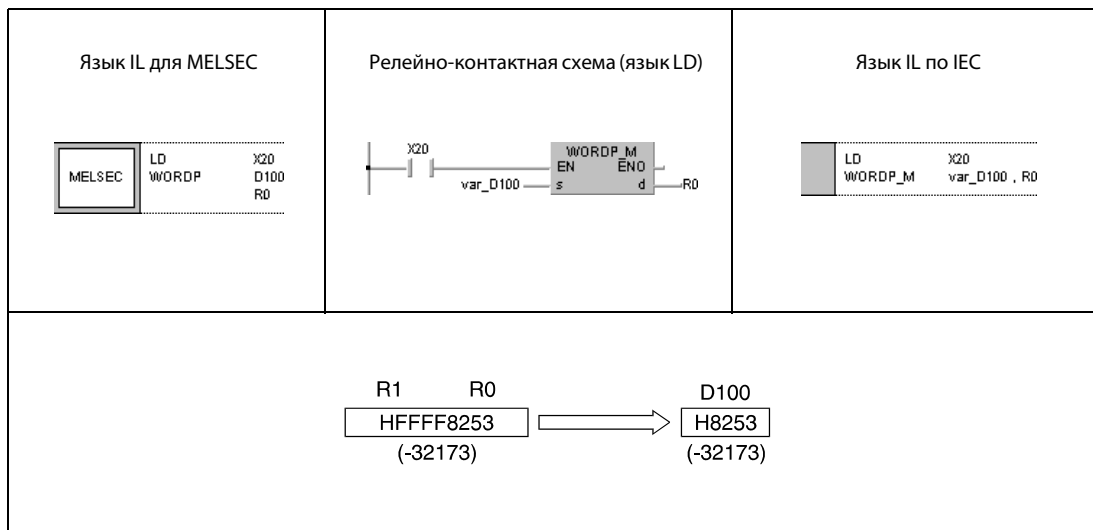
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в s и s + 1 значение находится вне диапазона от -32768 до 32767 (код ошибки 4100).

**Пример**

**WORDP**

Следующая программа при положительном фронте сигнала X20 преобразует 32-битное двоичное число, хранящееся в D100 и D101, в 16-битное двоичное число. Результат сохраняется в R0.



**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

6.3.7 GRY, GRYP, DGRY, DGRYP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

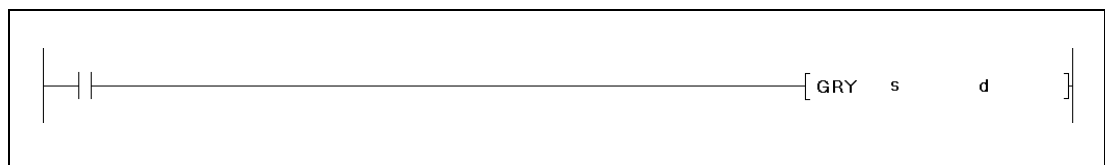
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

GX IEC Developer



GX Developer



Переменные

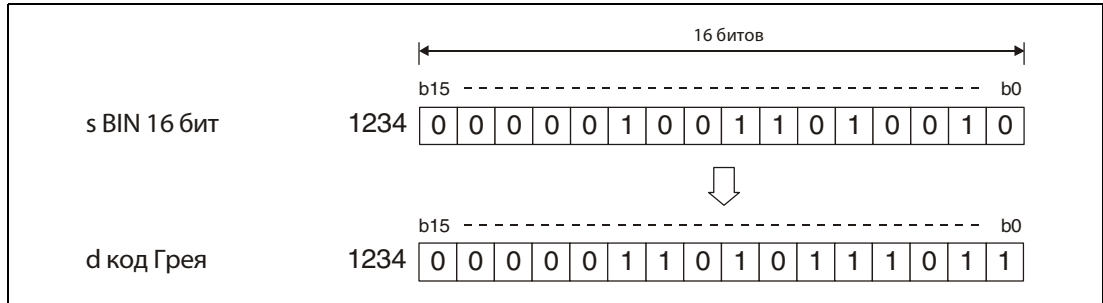
Операнд	Значение	Тип данных
s	Двоичные данные или операнд, в котором сохранены двоичные данные.	BIN, 16/32 бита
d	Операнд, в котором сохраняются конвертированные данные в коде Грея.	код Грея 16/32 бита

**Принцип действия**

**Преобразование 16- или 32-битных двоичных данных в код Грея**

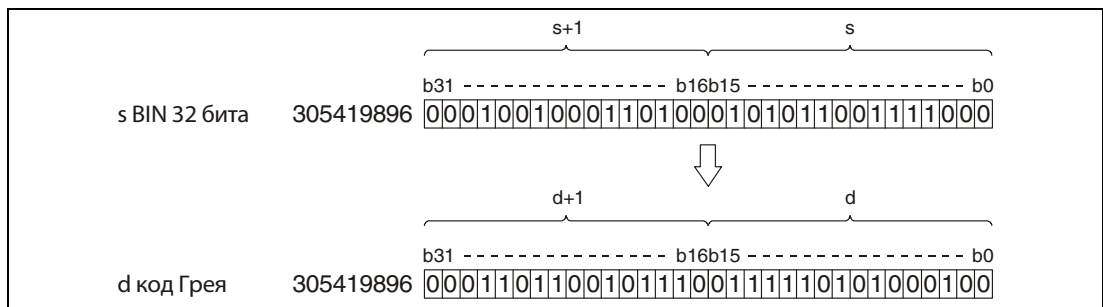
**GRY Преобразование 16-битных двоичных данных в код Грея**

16-битные двоичные данные, хранящиеся в s, кодируются в кодировке Грея. Результат сохраняется в d.



**DGRY Преобразование 32-битных двоичных данных в код Грея**

32-битные двоичные данные, сохраненные в s, кодируются в кодировке Грея. Результат сохраняется в d.



**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В s указано отрицательное значение.

**Пример 1** GRYP

Следующая программа при положительном фронте сигнала X10 преобразует хранящееся в D100 16-битное двоичное число в код Грея. Результат сохраняется в D200.



**Пример 2** DGRYP

Следующая программа при положительном фронте сигнала X1C преобразует 32-битное двоичное число, хранящееся в D10 и D11, в код Грея. Результат сохраняется в D100 и D101.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке ROU Пример 2 программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.



### 6.3.8 GBIN, GBINP, DGBIN, DGBINP

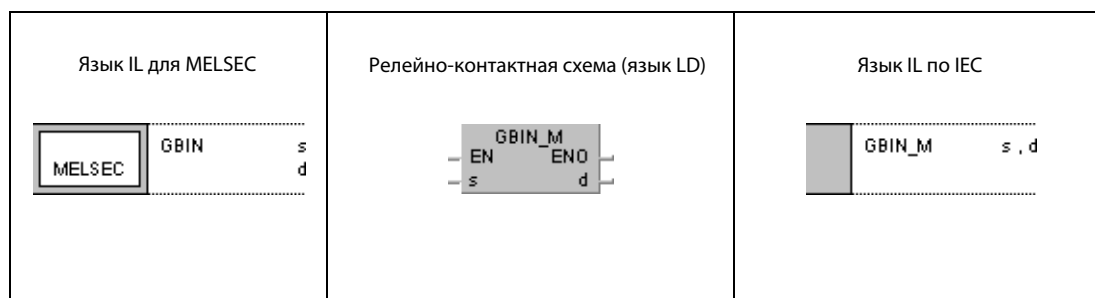
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

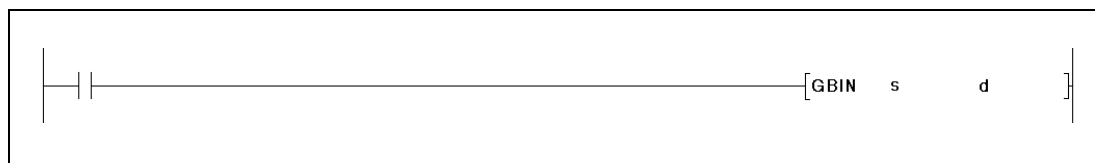
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

**GX IEC Developer**



**GX Developer**

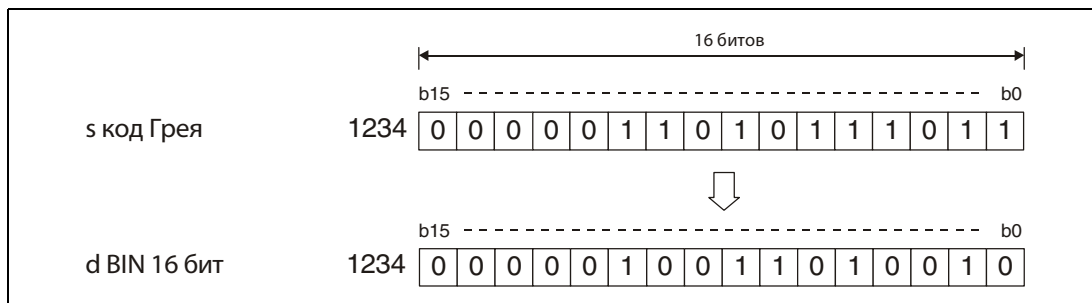


**Переменные**

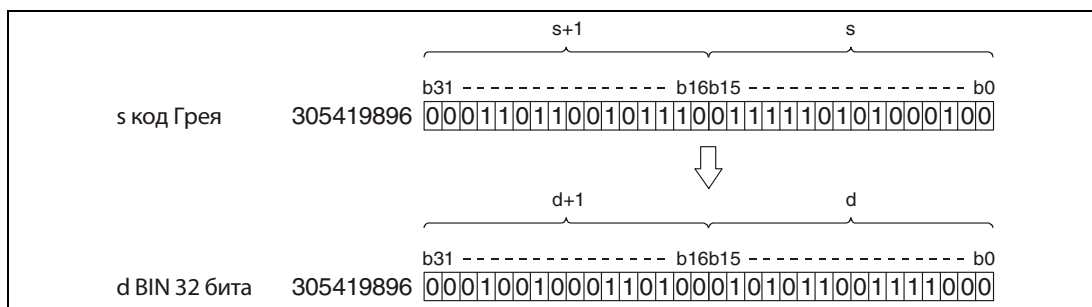
Операнд	Значение	Тип данных
s	Данные в кодировке Грея или операнд, в котором сохранены данные в кодировке Грея.	код Грея 16/32 бита
d	Операнд, в котором сохраняются конвертированные двоичные данные.	BIN, 16/32 бита

**Принцип действия****Преобразование данных в кодировке Грея в 16/32-битные двоичные данные****GBIN Преобразование данных в кодировке Грея в 16-битные двоичные данные**

Хранящиеся в *s* данные в кодировке Грея преобразуются в 16-битные двоичные данные. Результат сохраняется в *d*.

**DGBIN Преобразование данных в кодировке Грея в 32-битные двоичные данные**

Хранящиеся в *s* данные в кодировке Грея преобразуются в 32-битные двоичные данные. Результат сохраняется в *d*.

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- При применении команды GBIN указанное в *s* значение находится вне диапазона от 0 до 32767.
- При применении команды DGBIN указанное в *s* значение находится вне диапазона от 0 до 2147483647.

**Пример 1** GBINP

Следующая программа при положительном фронте сигнала X10 декодирует хранящиеся в D100 данные из кодировки Грея в 16-битное двоичное число и сохраняет результат в D200.

**Пример 2** DGBINP

Следующая программа при положительном фронте сигнала X1C преобразует хранящиеся в D10 и D11 данные в кодировке Грея в 32-битное двоичное число и сохраняет результат в D0 и D1.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU Пример 2 программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

6.3.9 NEG, NEGP, DNEG, DNEGP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

Операнды															Длина блока	Шагов	Индекс	Флаг							
битовые					словные (16 бит)					Константы		Указа-тели		Уро-вень				переноса	ошибки						
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z				V	K	H (16#)	P	I	N	M9012	M9010 M9011
d	●	●	●	●	●	●	●	●	●	●	●	●	●	●	2 <sup>1</sup>						K1 ↓ K4	3	●	●	●

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

<sup>2</sup> Команды DNEG и DNEGP не пригодны для процессоров AnN и AnS.

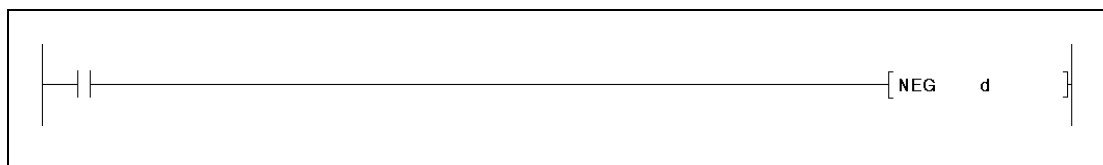
Операнды MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные							
d	●	●	●	●	●	●	—	—	—	—	2

GX IEC Developer



GX Developer



Переменные

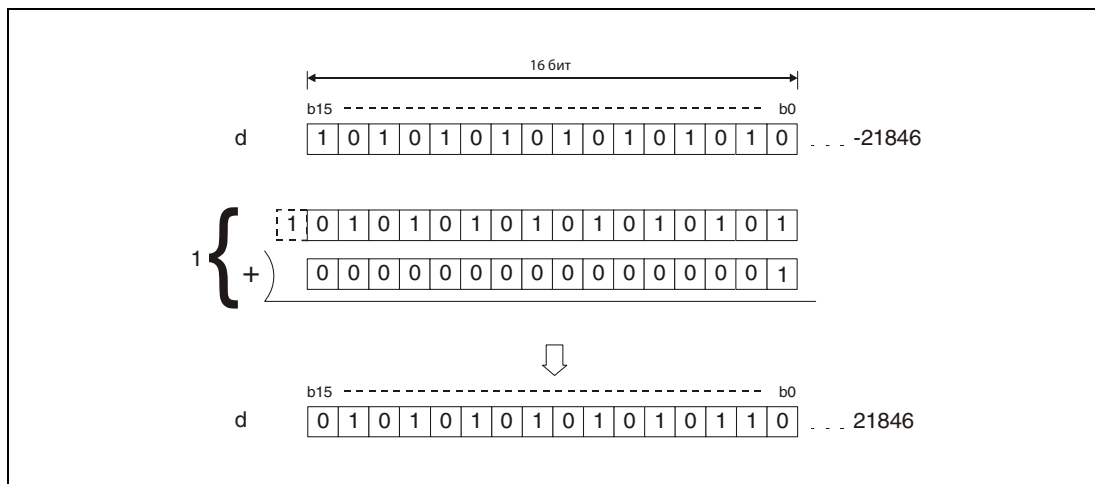
Операнд	Значение	Тип данных
d	Операнд, в котором хранятся двоичные данные, преобразуемые в дополнительный код.	BIN, 16/32 бита

**Принцип действия**

**Преобразование 16/32-битных двоичных данных в дополнительный код (изменение арифметического знака)**

**NEG Преобразование 16-битных двоичных данных в отрицательное число**

Функция NEG (преобразование в дополнительный код, называемая также логическим "НЕ") преобразует значение 16-битного слова данных в отрицательное. При этом 16-битное слово данных, хранящееся в d, сначала инвертируется, а в следующем шаге к нему прибавляется "1". Полученный дополнительный код снова сохраняется в d.

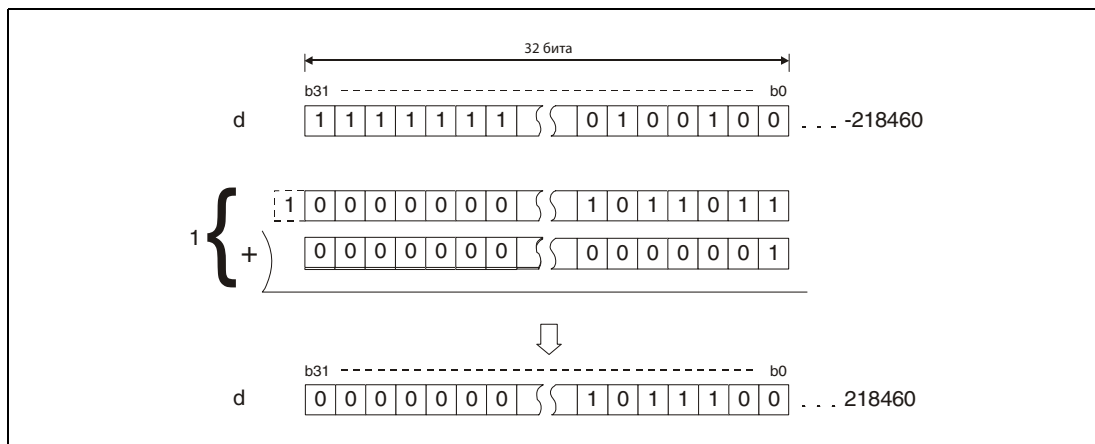


<sup>1</sup> инвертирование с последующим суммированием

Эта команда служит для преобразования отрицательного арифметического знака числа в положительный или положительного знака в отрицательный.

**DNEG Преобразование 32-битных двоичных данных в отрицательное число (только серии "Q" и "System Q")**

Функция DNEG (преобразование в дополнительный код, называемая также логическим "НЕ") преобразует значение 32-битного слова данных в отрицательное. Сначала 32-битное слово данных, содержащееся в d, инвертируется. В следующем шаге к нему прибавляется "1". Полученный дополнительный код снова сохраняется в d.

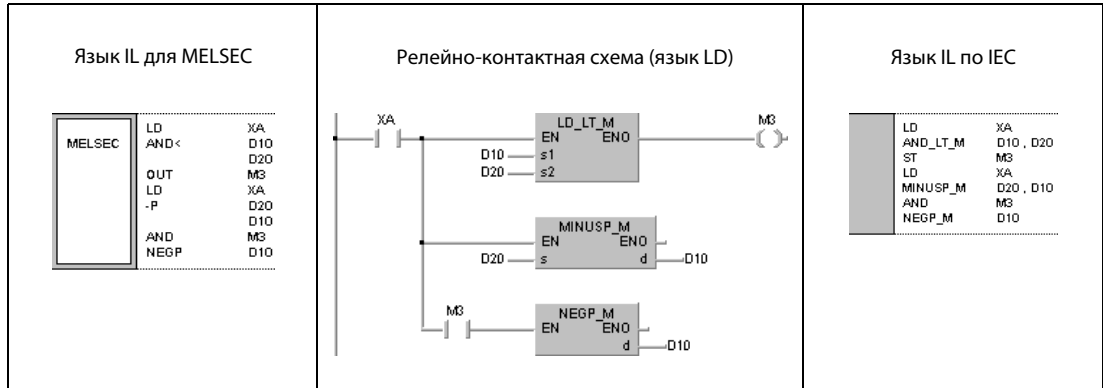


<sup>1</sup> инвертирование с последующим суммированием

**Пример**

NEGP

Следующая программа при положительном фронте сигнала XA вычитает значение в D10 из значения в D20. Если D10 меньше D20, перед этим устанавливается M3. Если результат вычитания "D10 – D20" отрицательный (M3 установлен), результат в D10 является абсолютным значением (дополнение до 2) и становится положительным.



### 6.3.10 ENEG, ENEG

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

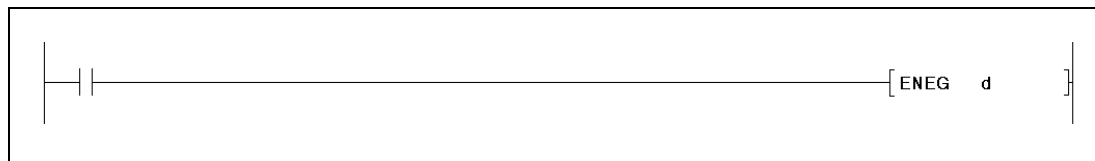
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
d	—	●	●	—	●	●	—	—	—	—	2

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
d	Операнд с числом с плавающей запятой, у которого требуется изменить арифметический знак.	REAL

**Принцип действия**

**Изменение арифметического знака чисел с плавающей запятой**

**ENEG    Изменение арифметического знака чисел с плавающей запятой**

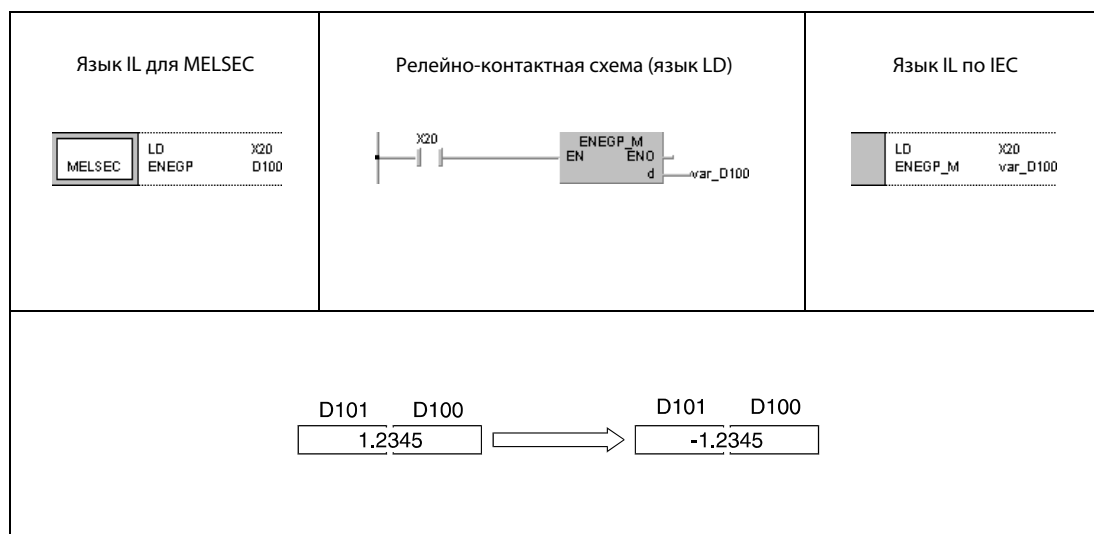
Это команда изменяет знак хранящегося в d числа с плавающей запятой. Результат сохраняется снова в d.

Эта команда служит для преобразования отрицательного арифметического знака числа в положительный или положительного знака в отрицательный.

**Пример**

ENEGP

Следующая программа при положительном фронте сигнала X20 изменяет знак числа с плавающей запятой, хранящегося в D100 и D101, и снова сохраняет результат в D100 и D101.



**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.



**6.3.11 BKBCD, BKBCDP**

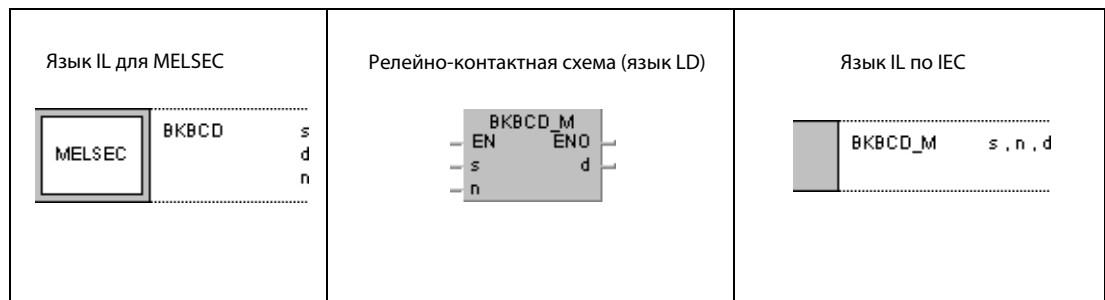
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

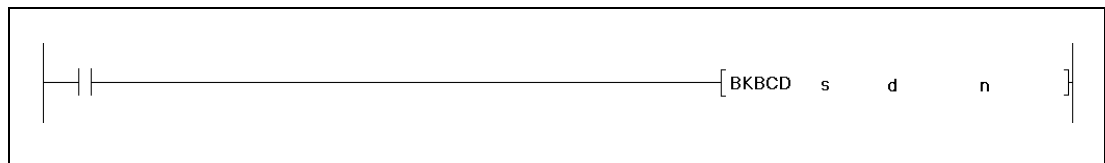
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	—	SM0	4
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Операнд, в котором сохранены конвертируемые двоичные блоки данных.	BIN, 16 бит
d	Операнд, в котором сохраняются конвертированные блоки данных типа BCD.	
n	количество конвертируемых блоков данных	

**Принцип действия**

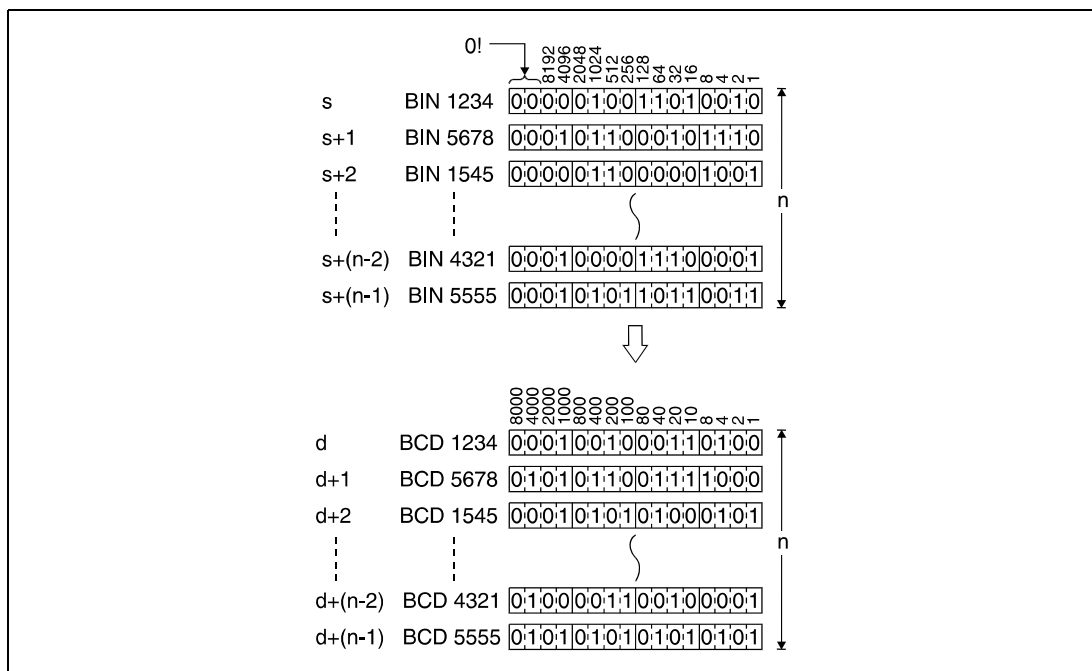
**Поблочное преобразование двоичных данных в двоично-десятичные**

**ВКBCD Преобразование блоков 16-битных двоичных данных в блоки 4-значных двоично-десятичных данных**

Эта команда преобразует n-ый 16-битный блок данных, хранящихся в s, и сохраняет соответствующий преобразованный n-ый блок 4-значных двоично-десятичных данных в d.

Указанные в s блоки двоичных данных должны иметь значение в диапазоне между 0 и 9999.

Два старших бита блоков 16-битных двоичных данных должны иметь состояние "0".



**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

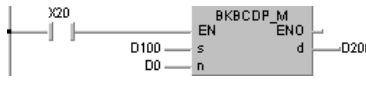

- Указанное в n количество блоков данных в s и d находится вне области операндов, предусмотренной для сохранения (код ошибки 4101).
- Указанные в s двоичные данные находятся вне диапазона между 0 и 9999 (код ошибки 4100).
- Области памяти s и d перекрываются (код ошибки 4101).

Информация о применении индексированной адресации имеется в разделе 3.6.

**Пример**

**ВКBCDP**

Следующая программа при положительном фронте сигнала X20 преобразует блоки 16-битных двоичных данных, хранящиеся начиная с D100, в блоки 4-значных двоично-десятичных данных. Блоки данных типа BCD сохраняются, начиная с D200. Количество конвертируемых блоков данных (3) указано в D0.

Язык IL для MELSEC	Релейно-контактная схема (язык LD)	Язык IL по IEC																																																																																	
<pre> MELSEC LD X20 BKBCDP D100 D200 D0                     </pre>		<pre> IEC LD X20 BKBCDP_M D100, D0, D200                     </pre>																																																																																	
<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: left;"> <table border="0"> <tr><td></td><td></td><td style="text-align: right;">8192</td></tr> <tr><td></td><td></td><td style="text-align: right;">4096</td></tr> <tr><td></td><td></td><td style="text-align: right;">2048</td></tr> <tr><td></td><td></td><td style="text-align: right;">1024</td></tr> <tr><td></td><td></td><td style="text-align: right;">512</td></tr> <tr><td></td><td></td><td style="text-align: right;">256</td></tr> <tr><td></td><td></td><td style="text-align: right;">128</td></tr> <tr><td></td><td></td><td style="text-align: right;">64</td></tr> <tr><td></td><td></td><td style="text-align: right;">32</td></tr> <tr><td></td><td></td><td style="text-align: right;">16</td></tr> <tr><td></td><td></td><td style="text-align: right;">8</td></tr> <tr><td></td><td></td><td style="text-align: right;">4</td></tr> <tr><td></td><td></td><td style="text-align: right;">2</td></tr> <tr><td></td><td></td><td style="text-align: right;">1</td></tr> </table> <p>D100 BIN 5432 00001010100111000</p> <p>D101 BIN 4444 0001000101011100</p> <p>D102 BIN 3210 0000110010001010</p> </div> <div style="text-align: center;">  </div> <div style="text-align: right;"> <p>D0 3</p> <table border="0"> <tr><td></td><td></td><td style="text-align: right;">8000</td></tr> <tr><td></td><td></td><td style="text-align: right;">4000</td></tr> <tr><td></td><td></td><td style="text-align: right;">2000</td></tr> <tr><td></td><td></td><td style="text-align: right;">1000</td></tr> <tr><td></td><td></td><td style="text-align: right;">500</td></tr> <tr><td></td><td></td><td style="text-align: right;">250</td></tr> <tr><td></td><td></td><td style="text-align: right;">125</td></tr> <tr><td></td><td></td><td style="text-align: right;">62</td></tr> <tr><td></td><td></td><td style="text-align: right;">31</td></tr> <tr><td></td><td></td><td style="text-align: right;">15</td></tr> <tr><td></td><td></td><td style="text-align: right;">7</td></tr> <tr><td></td><td></td><td style="text-align: right;">3</td></tr> <tr><td></td><td></td><td style="text-align: right;">1</td></tr> </table> <p>D200 BCD 5432 0101010000110010</p> <p>D201 BCD 4444 0100010001000100</p> <p>D202 BCD 3210 0011001000010000</p> </div> </div>					8192			4096			2048			1024			512			256			128			64			32			16			8			4			2			1			8000			4000			2000			1000			500			250			125			62			31			15			7			3			1
		8192																																																																																	
		4096																																																																																	
		2048																																																																																	
		1024																																																																																	
		512																																																																																	
		256																																																																																	
		128																																																																																	
		64																																																																																	
		32																																																																																	
		16																																																																																	
		8																																																																																	
		4																																																																																	
		2																																																																																	
		1																																																																																	
		8000																																																																																	
		4000																																																																																	
		2000																																																																																	
		1000																																																																																	
		500																																																																																	
		250																																																																																	
		125																																																																																	
		62																																																																																	
		31																																																																																	
		15																																																																																	
		7																																																																																	
		3																																																																																	
		1																																																																																	

6.3.12 ВКВІН, ВКВІП

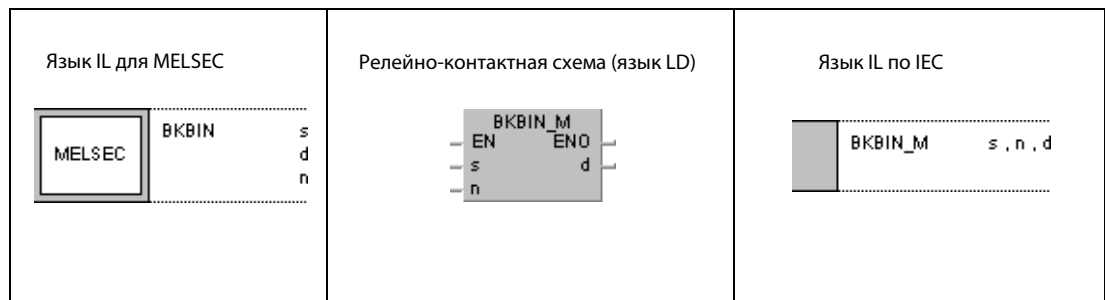
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

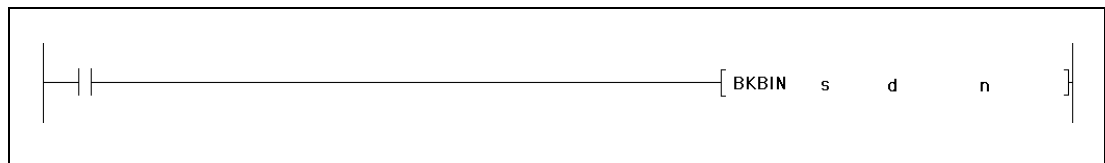
Операнды MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	—	SM0	4
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer



GX Developer



Переменные

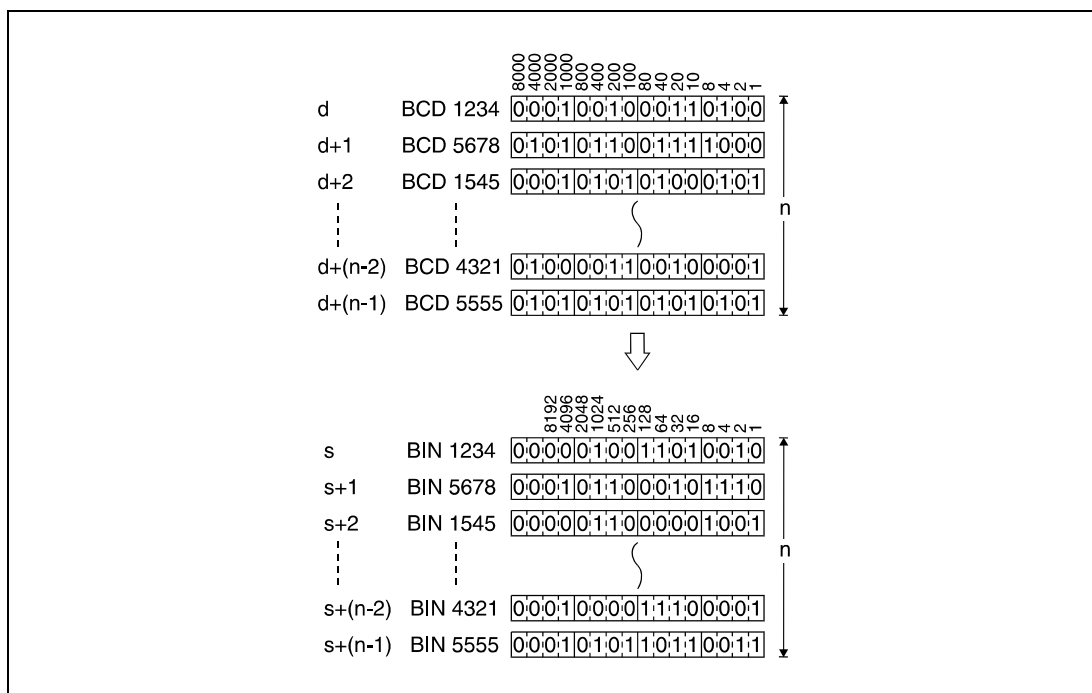
Операнд	Значение	Тип данных
s	Операнд, в котором сохранены конвертируемые блоки данных типа BCD.	BIN, 16 бит
d	Операнд, в котором сохраняются конвертированные блоки двоичных данных.	
n	количество конвертируемых блоков данных	

**Принцип действия**

**Поблочное преобразование двоично-десятичных данных в двоичные ВКВІВ, ВКВІР**  
**Преобразование блоков 4-значных двоично-десятичных данных в блоки 16-битных двоичных данных**

Эта команда преобразует n-ый блок 4-значных двоично-десятичных данных, хранящихся в s, и сохраняет преобразованный n-ый блок 16-битных двоичных данных в d.

Указанные в s блоки двоично-десятичных данных должны иметь значения в диапазоне между 0 и 9999.



**Источники ошибок**

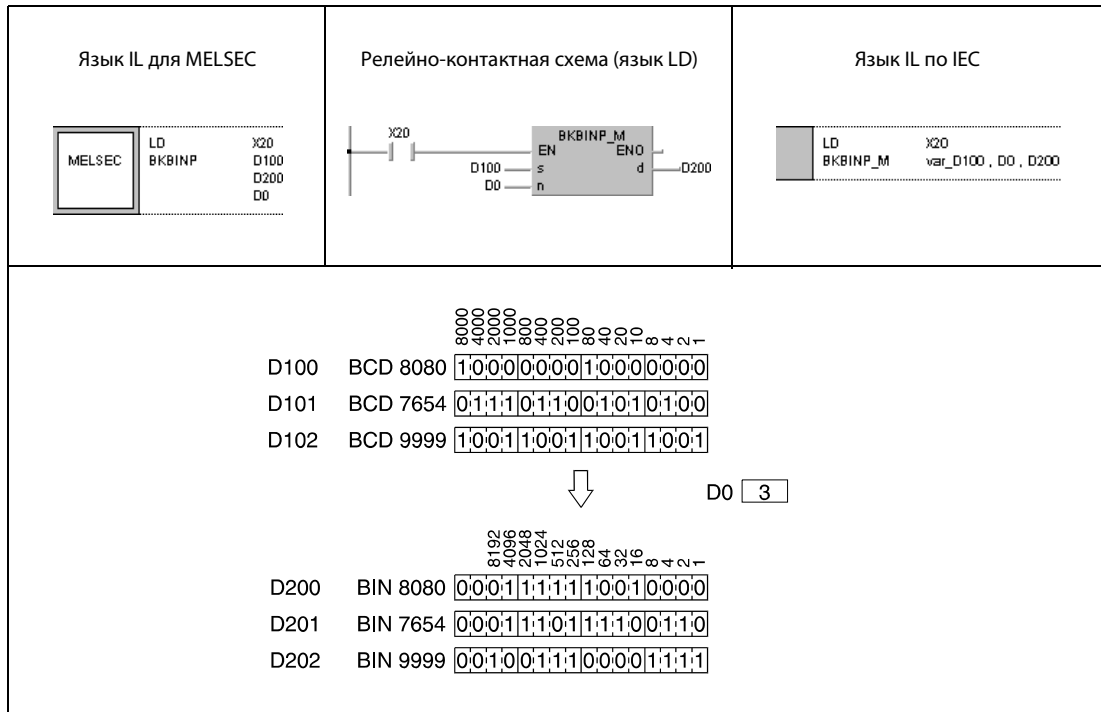
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в n количество блоков данных в s и d находится вне области операндов, предусмотренной для сохранения (код ошибки 4101).
- Указанные в s двоично-десятичные данные находятся вне диапазона между 0 и 9999.
- Области памяти s и d перекрываются.

Информация о применении индексированной адресации имеется в разделе 3.6 этого руководства.

**Пример** ВКВІНР

Следующая программа при положительном фронте сигнала X20 преобразует блоки 4-значных двоично-десятичных данных, хранящиеся начиная с D100, в блоки 16-битных двоичных данных. Блоки двоичных данных сохраняются, начиная с D200. Количество конвертируемых блоков данных (3) указано в D0.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

## 6.4 Команды передачи

Команды передачи данных позволяют переносить данные, менять данные местами или инвертировать данные. В общей сложности имеется 24 различных команды. Обзорный перечень этих команд дан в таблице ниже.

**ПРИМЕЧАНИЕ** *Переданные данные остаются сохраненными до тех пор, пока они не будут заменены. По этой причине данные остаются сохраненными даже в том случае, если входное условие команды передачи более не выполняется.*

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC
Передача (16/32 бита)	MOV	MOV_M
	MOVP	MOVP_M
	DMOV	DMOV_M
	DMOVP	DMOVP_M
Передача чисел с плавающей запятой	EMOV	EMOV_M
	EMOVP	EMOVP_M
Передача строковых величин	\$MOV	STRING_MOV_M
	\$MOVP	STRING_MOVP_M
Инверсная передача (16/32 бита)	CML	CML_M
	CMLP	CMLP_M
	DCML	DCML_M
	DCMLP	DCMLP_M
Поблочная передача	BMOV	BMOV_M
	BMOVP	BMOVP_M
Блочная передача идентичных данных	FMOV	FMOV_M
	FMOVP	FMOVP_M
Обмен (16/32 бита)	XCH	XCH_M
	XCHP	XCHP_M
	DXCH	DXCH_M
	DXCHP	DXCHP_M
Обмен (16-битных блоков)	BXCH	BXCH_M
	BXCHP	BXCHP_M
Переставление (старшего и младшего байта)	SWAP	SWAP_MD
	SWAPP	SWAP_P_MD

**ПРИМЕЧАНИЕ** *В редакторах по IEC используйте стандартные команды IEC.*

6.4.1 MOV, MOVP, DMOV, DMOVP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

Операнды															Длина блока	Шагов	Индекс	Флаг					
битовые					словные (16 бит)					Константы		Указатели		Уровень				переноса	ошибки				
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z				V	K	H (16#)	P	I	N
<b>MOV</b>																							
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●								●
<b>DMOV</b>																							
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							●

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

Операнды MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□NG□□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные							
s	●	●	●	●	●	●	●	—	—	—	3 <sup>1)</sup>
d	●	●	●	●	●	●	—	—	—	—	3 <sup>2)</sup>

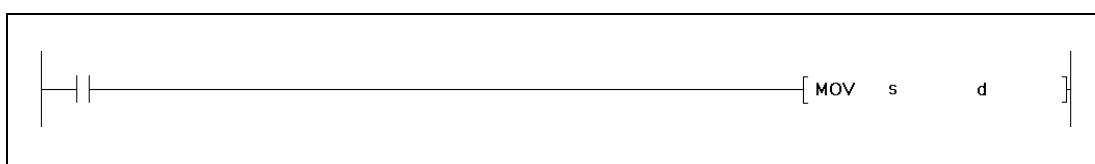
<sup>1</sup> Количество шагов программы зависит от типа центрального процессора и используемых операндов.  
 При использовании процессоров QnA или однопроцессорного модуля ЦП серии "Q": 3  
 В случае многопроцессорных ЦПУ серии "Q" и внутренних словных операндов (кроме регистра файлов ZR) или констант: 2  
 В случае многопроцессорного ЦПУ серии "Q" и битовых операндов с адресом, кратным 16, имеющих обозначение битового блока 4 и не индексируемых: 2  
 При использовании многопроцессорного модуля ЦП серии "Q" и иных операндов кроме вышеуказанных: 3

<sup>2</sup> Количество шагов программы зависит от типа центрального процессора.  
 В случае процессора QnA или многопроцессорного модуля ЦП серии "Q": 3  
 В случае однопроцессорного модуля ЦП серии "Q": 2

GX IEC Developer



GX Developer





**Переменные**

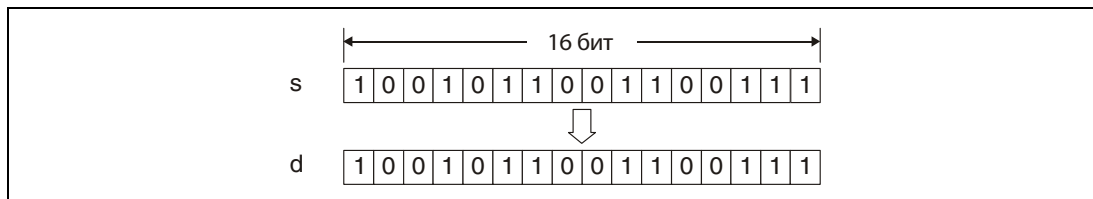
Операнд	Значение	Тип данных
s	Двоичные данные или операнд, в котором сохранены двоичные данные.	BIN, 16/32 бита
d	Операнд, в который передаются двоичные данные.	

**Принцип действия**

**Передача 16/32-битных двоичных данных**

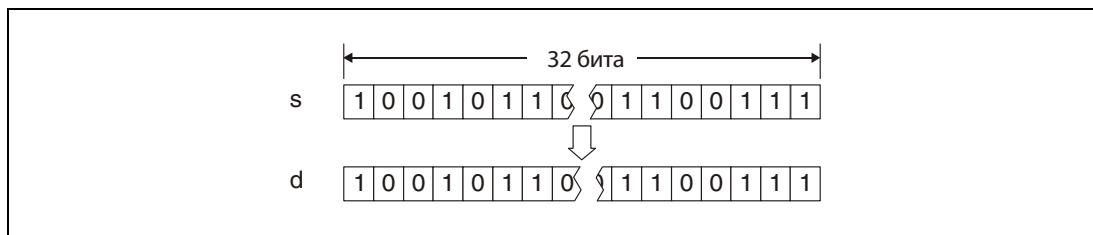
**MOV Передача данных (16 бит)**

Команда MOV передает 16-битные данные, указанные в s, в операнд, указанный в d.



**DMOV Передача данных (32 бита)**

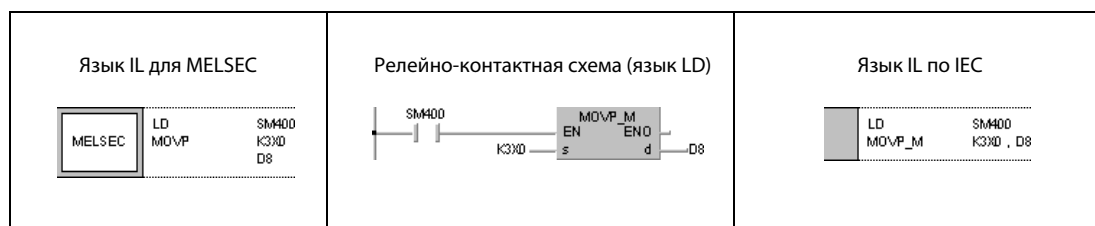
Команда DMOV передает 32-битные данные, указанные в s, в операнд, указанный в d.



**Пример 1**

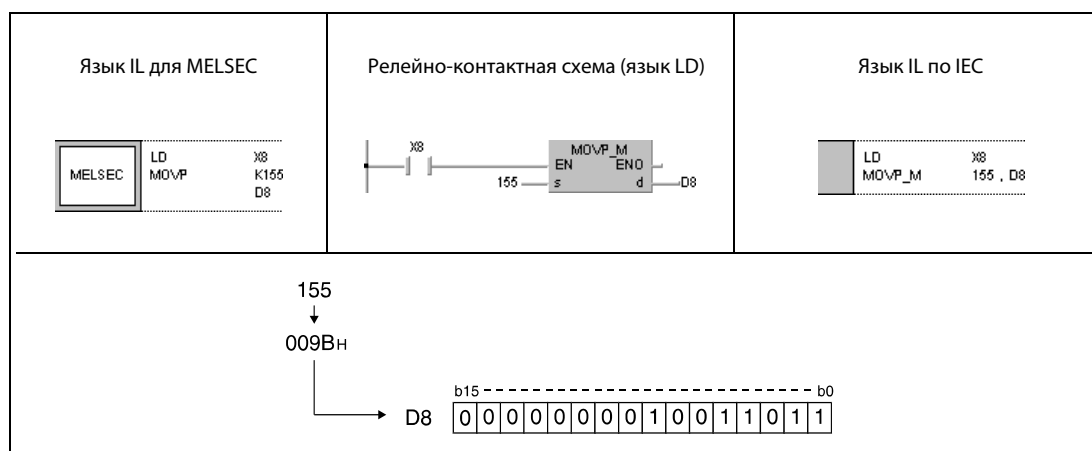
**MOVP**

Следующая программа при положительном фронте сигнала SM400 передает данные из X0...XB в D8.



**Пример 2** MOVP

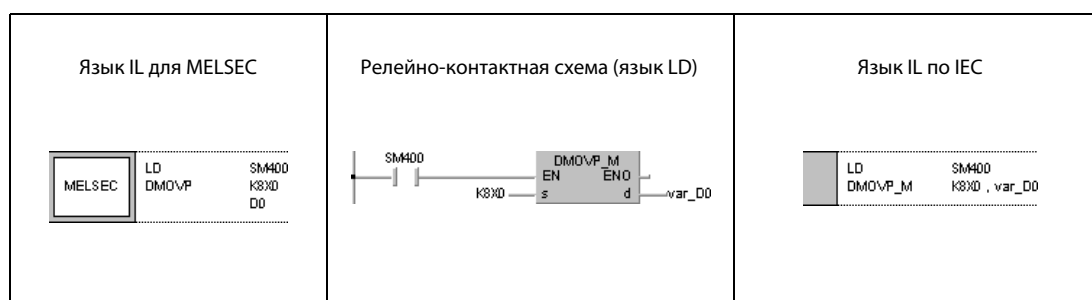
Следующая программа при положительном фронте сигнала X8 записывает значение 155 в регистр D8 в двоичном виде.

**Пример 3** DMOVP

Следующая программа при положительном фронте сигнала SM400 передает данные из регистров D0 и D1 в регистры D7 и D8.

**Пример 4** DMOVP

Следующая программа при положительном фронте сигнала SM400 передает данные из X0...X1F в D0 и D1.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU примеры программ 3 и 4 не работоспособны. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию можно найти в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

### 6.4.2 EMOV, EMOVP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

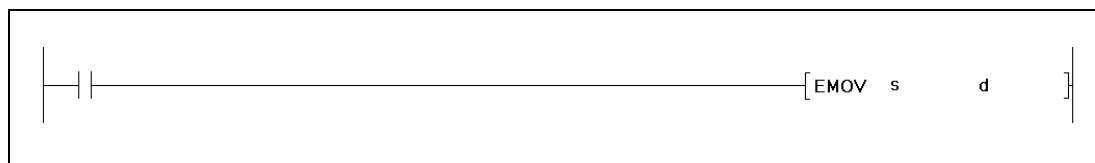
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы E			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	●	●	—	●	—	3	
d	—	●	●	—	●	●	—	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

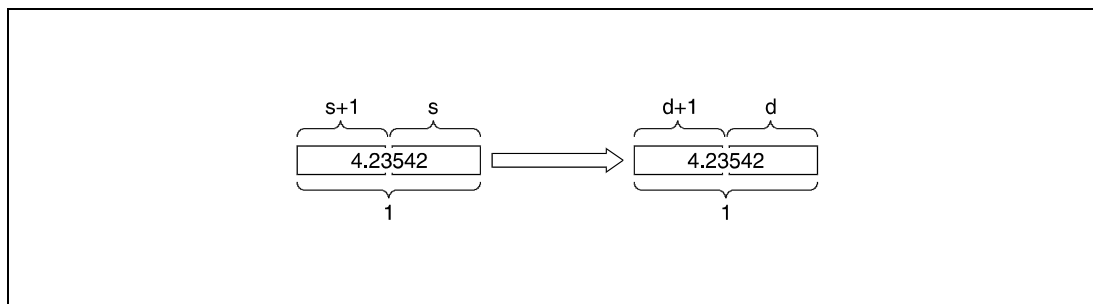
Операнд	Значение	Тип данных
s	Число с плавающей запятой или операнд, в котором сохранено передаваемое число с плавающей запятой.	REAL
d	Операнд, в который передается число с плавающей запятой.	

**Принцип действия**

**Передача чисел с плавающей запятой**

**EMOV/EMOVP Передача чисел с плавающей запятой**

Команда EMOV передает число с плавающей запятой, указанное в s, в операнд, указанный в d.

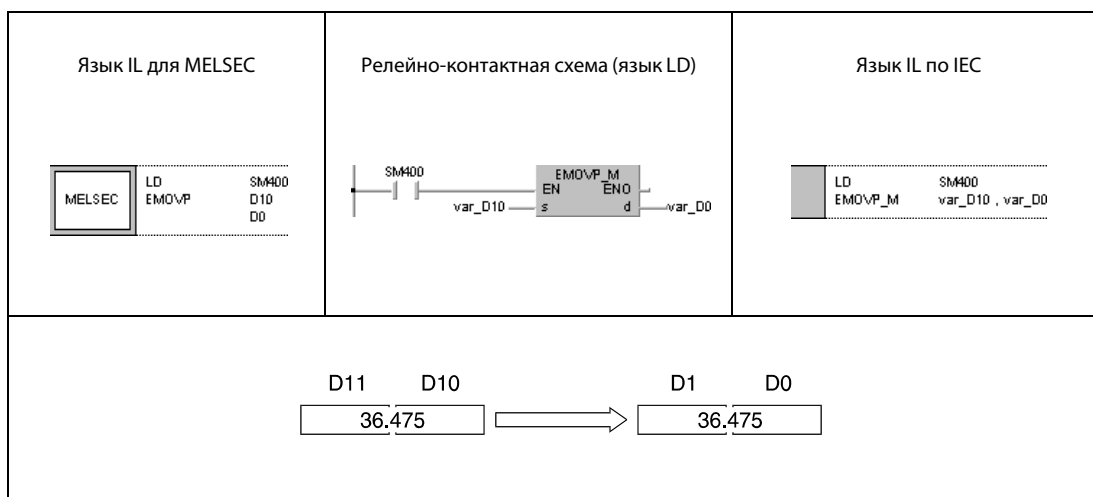


<sup>1</sup> число с плавающей запятой (данные типа REAL)

**Пример 1**

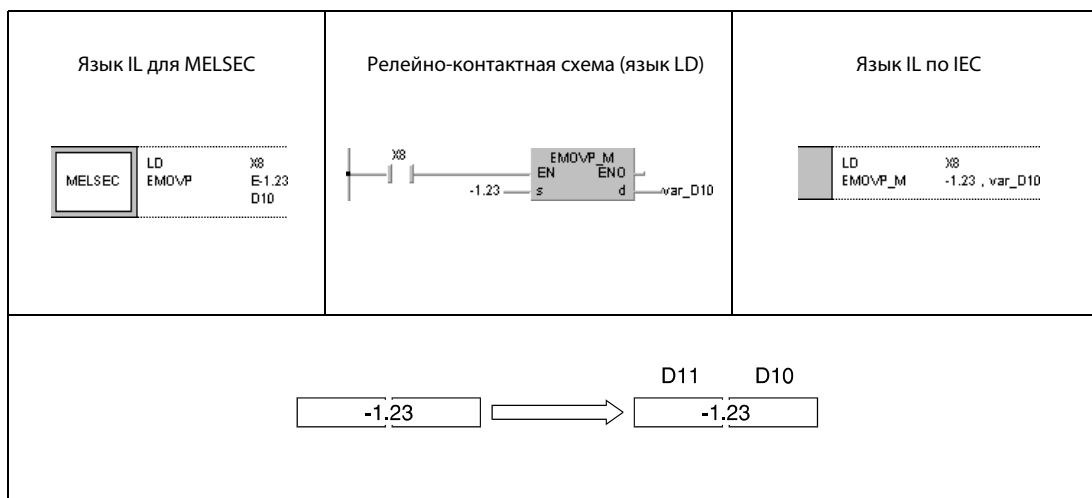
**EMOVP**

Следующая программа при положительном фронте сигнала SM400 передает число с плавающей запятой, хранящееся в регистрах D10 и D11, в регистры D0 и D1.



**Пример 2** EMOVP

Следующая программа при положительном фронте сигнала X8 передает число типа REAL –1.23 в регистры D10 и D11.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

### 6.4.3 \$MOV, \$MOVP

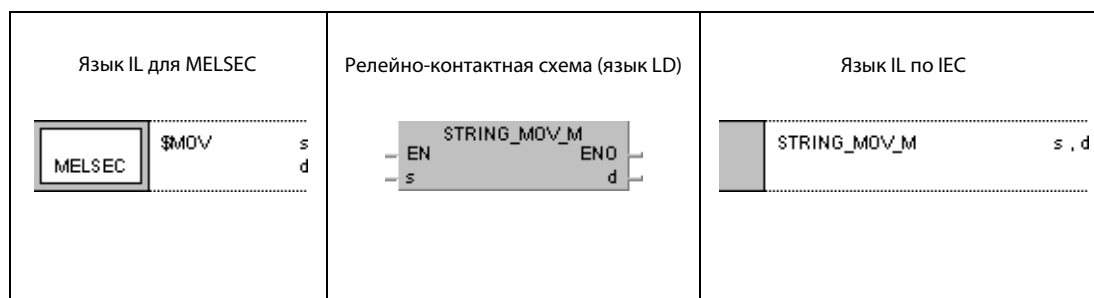
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

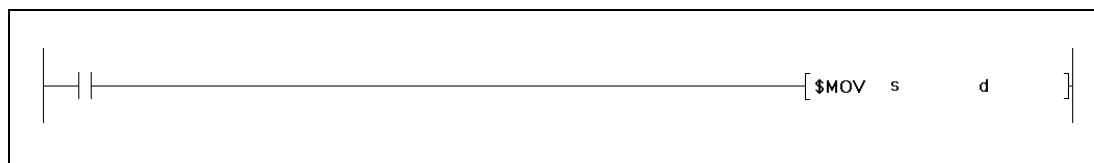
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□\G□□	Индексные регистры Zn	Константы \$			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	●	—	SM0	3
d	—	●	●	—	—	—	—	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

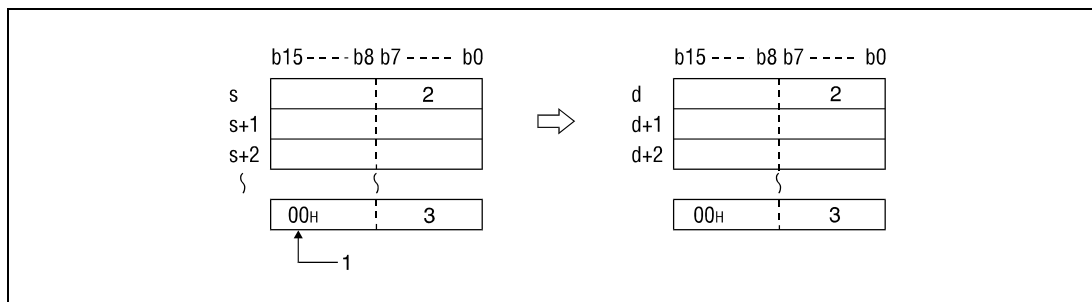
Операнд	Значение	Тип данных
s	Строковая величина или операнд, в котором сохранена передаваемая строковая величина.	строковая величина
d	Операнд, в который передается строковая величина.	

**Принцип действия**

**Передача строковых величин**

**\$MOV Передача строковых величин**

Команда \$MOV передает хранящиеся в s байты строковой величины в d. При этом за одну операцию передается вся строковая величина, начиная с первого знака (байта) до байта, содержащего код "00н" (конец строковой величины).

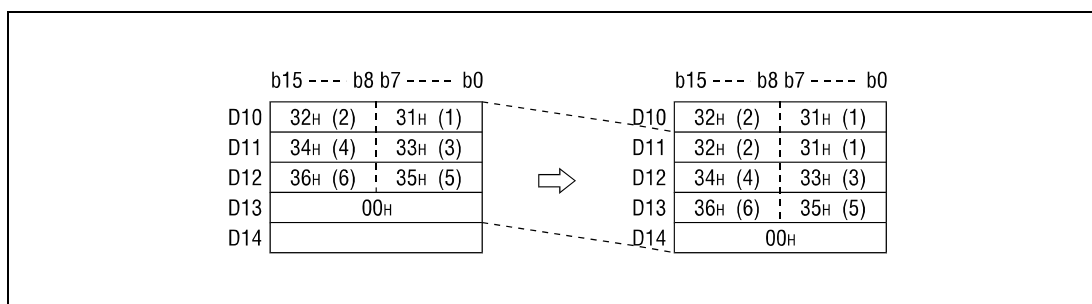


<sup>1</sup> Обозначает конец строковой величины.

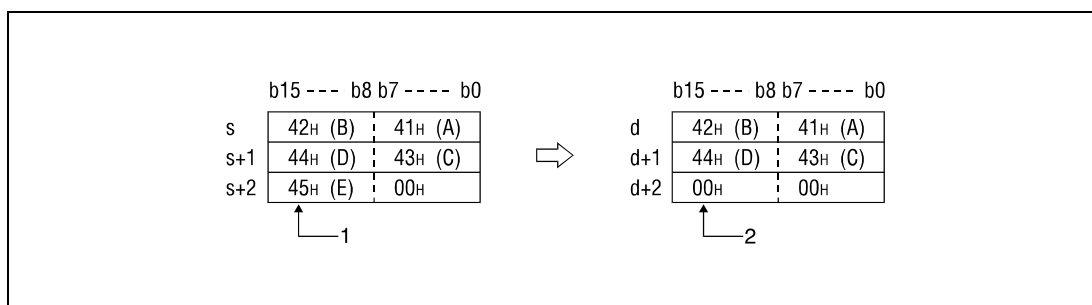
<sup>2</sup> 1-й знак

<sup>3</sup> n-ный знак

Команда \$MOV выполняется без сообщения об ошибке даже в том случае, если области передаваемых данных s...s + n перекрываются с областями d...d + n, предусмотренными для сохранения. Если хранящаяся в D10...D13 строковая величина передается в D11...D14, образуется следующий результат.



Если в s + n перед знаком старшего байта находится код "00н", то следующий за этим кодом знак при передаче не учитывается и в соответствующий байт в d + n также записывается код "00н".



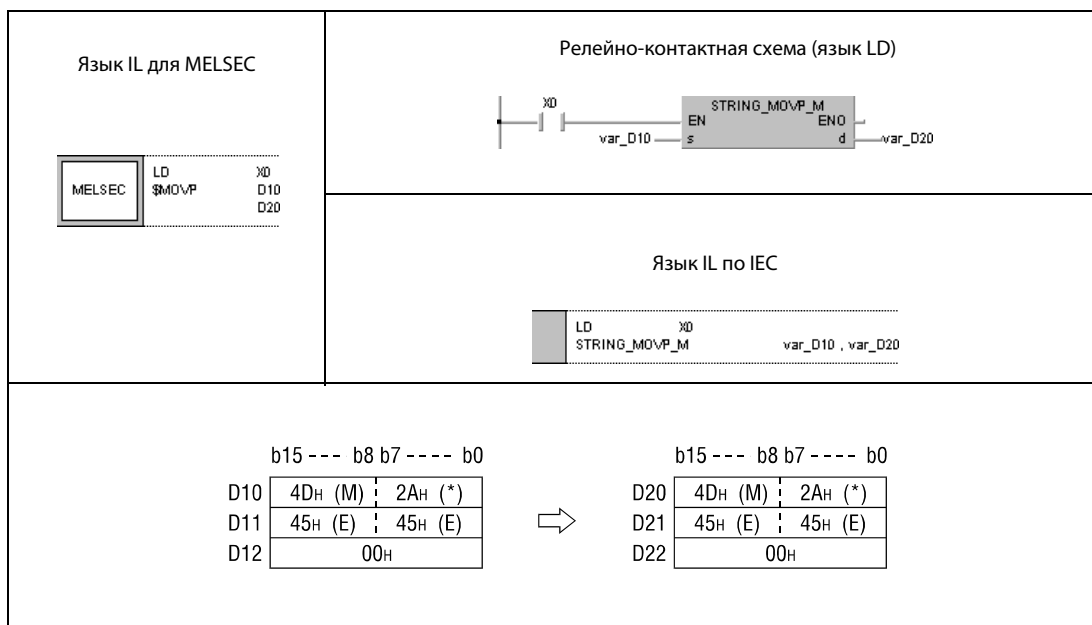
<sup>1</sup> Этот знак не передается.

<sup>2</sup> Этот байт автоматически записывается код "00н".

**Источники ошибок** В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В строковой величине, указанной в s, нет кода "00н" (код ошибки 4101).
- Строковая величина не может быть передана в d полностью.

**Пример** Следующая программа при положительном фронте сигнала X0 передает строковую величину, хранящуюся в регистрах D10...D12, в регистры D20...D22.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.



### 6.4.4 CML, CMLP, DCML, DCMLP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

Операнды																Длина блока	Шагов	Индекс	Флаг				
битовые						словные (16 бит)						Константы		Указа-тели					Уро-вень		пере-носа	ошиб-ки	
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V				K	H (16#)	P	I	N
CML																K1 ↓ K4	5	●					
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						●	●	
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●								
DCML																K1 ↓ K8	7	●					
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						●	●	
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●								

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

**Операнды MELSEC Q**

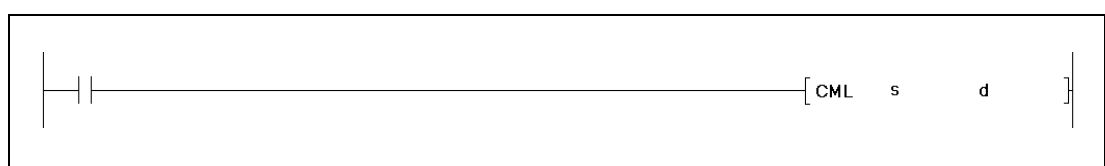
Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□G□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные							
s	●	●	●	●	●	●	●	—	—	—	3 <sup>1)</sup>
d	●	●	●	●	●	●	—	—	—	—	

<sup>1</sup> Количество шагов программы зависит от типа центрального процессора и используемых операндов.  
 В случае процессора QnA или однопроцессорного модуля ЦП серии "Q": 3  
 В случае многопроцессорного ЦПУ серии "Q" и внутренних словных операндов (кроме регистра файлов ZR) или констант: 2  
 В случае многопроцессорного ЦПУ серии "Q" и битовых операндов с адресом, кратным 16, имеющих обозначение битового блока K4 и не индексируемых: 2  
 При использовании многопроцессорного модуля ЦП серии "Q" и иных операндов кроме вышеуказанных: 3

**GX IEC Developer**



**GX Developer**



**Переменные**

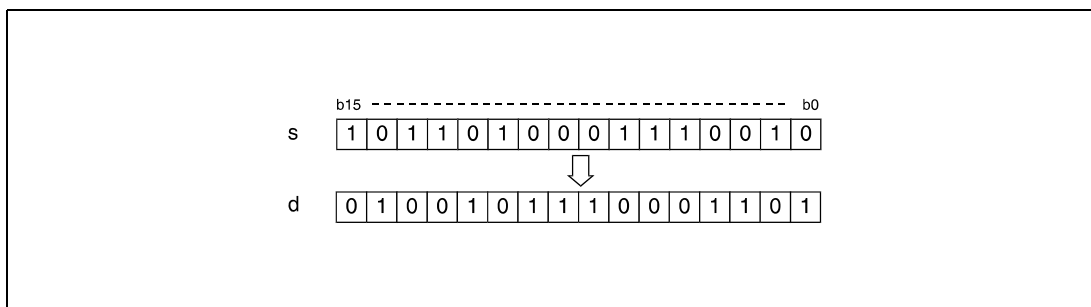
Операнд	Значение	Тип данных
s	Двоичные данные или операнд, в котором сохранены инвертируемые двоичные данные.	BIN, 16/32 бита
d	Операнд, в котором сохраняются инвертированные двоичные данные.	

**Принцип действия**

**Инвертирование 16/32-битных двоичных данных**

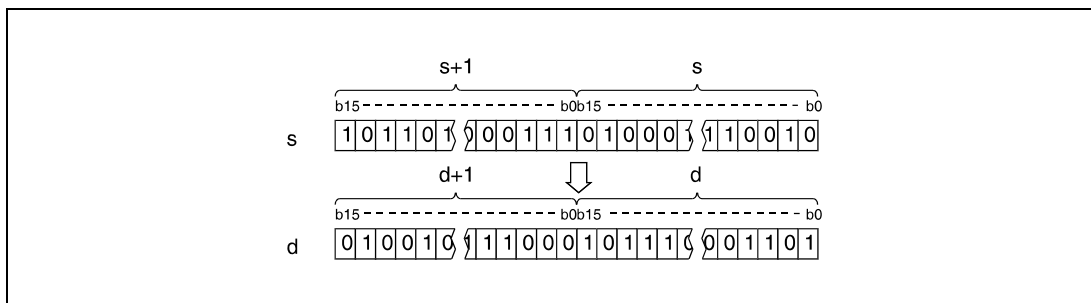
**CML Инвертирование данных (16 бит)**

Команда CML побитно инвертирует 16-битное значение, указанное в s. Результат записывается в d.



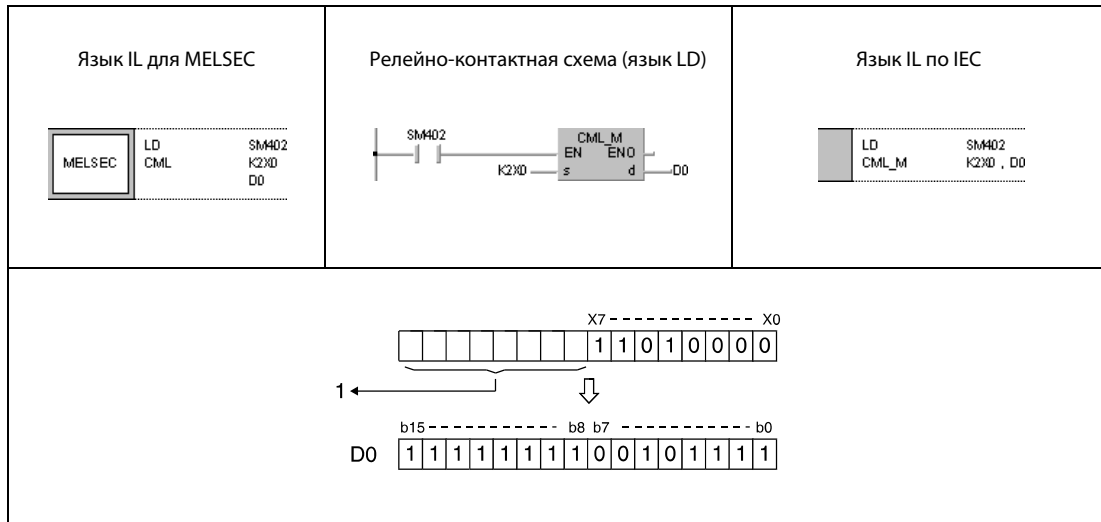
**DCML Инвертирование данных (32 бита)**

Команда DCML побитно инвертирует 32-битное значение данных, указанное в s. Результат записывается в d.



**Пример 1** CML

Следующая программа при включенном маркере SM402 передает данные X0...X7 в инвертированном виде в D0.

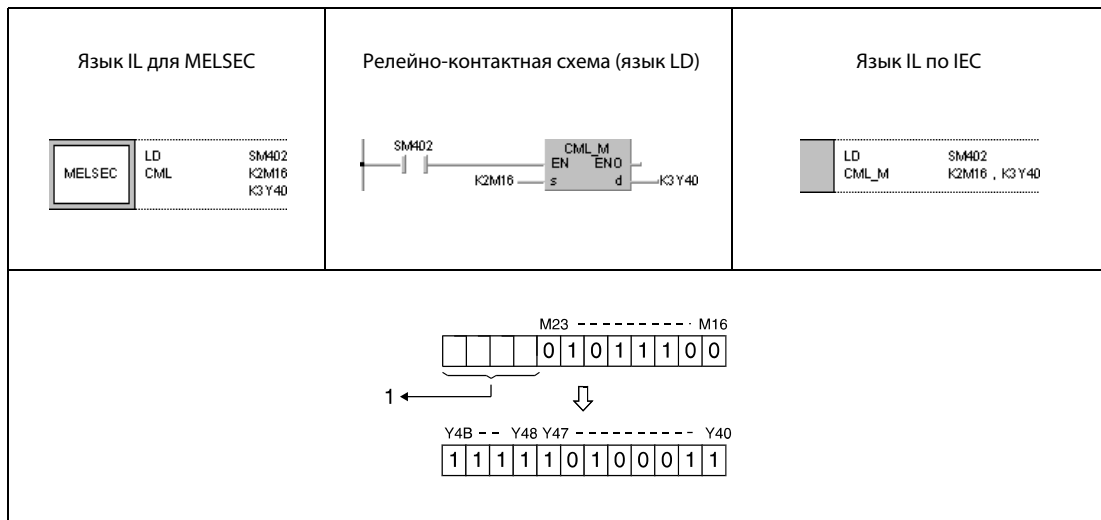


<sup>1</sup> Не указанные биты считываются в виде нулей.

Учитывайте, что в s должно быть меньше битов, чем в d.

**Пример 2** CML

Следующая программа при включенном маркере SM402 передает данные маркеров M16...M23 в инвертированном виде в блок K3 адреса Y40 (Y40...Y4F). Биты Y48...Y4B устанавливаются на "1", так как для них считан "0".

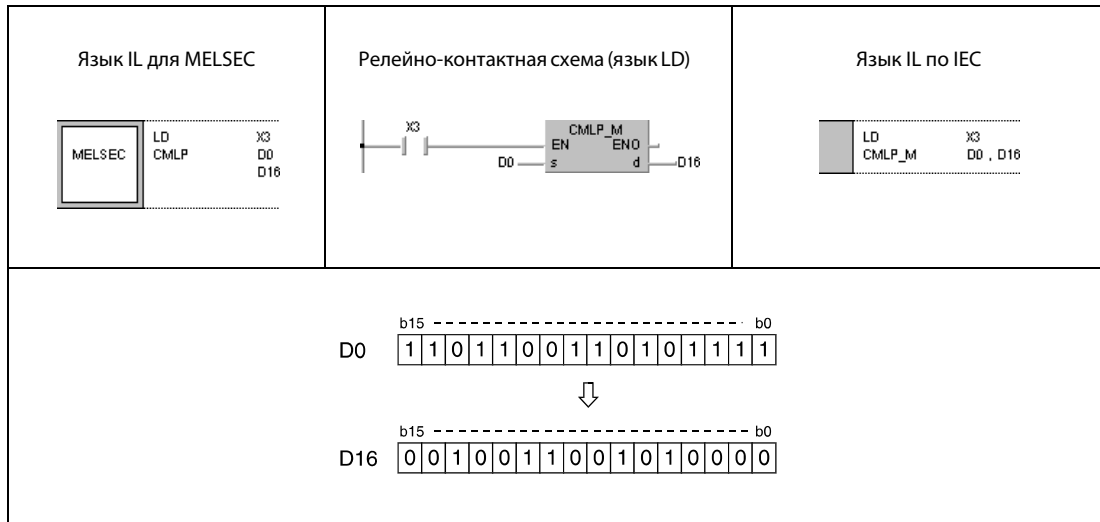


<sup>1</sup> Не указанные биты считываются в виде нулей.

Количество битов в s должно быть меньше, чем количество битов в d.

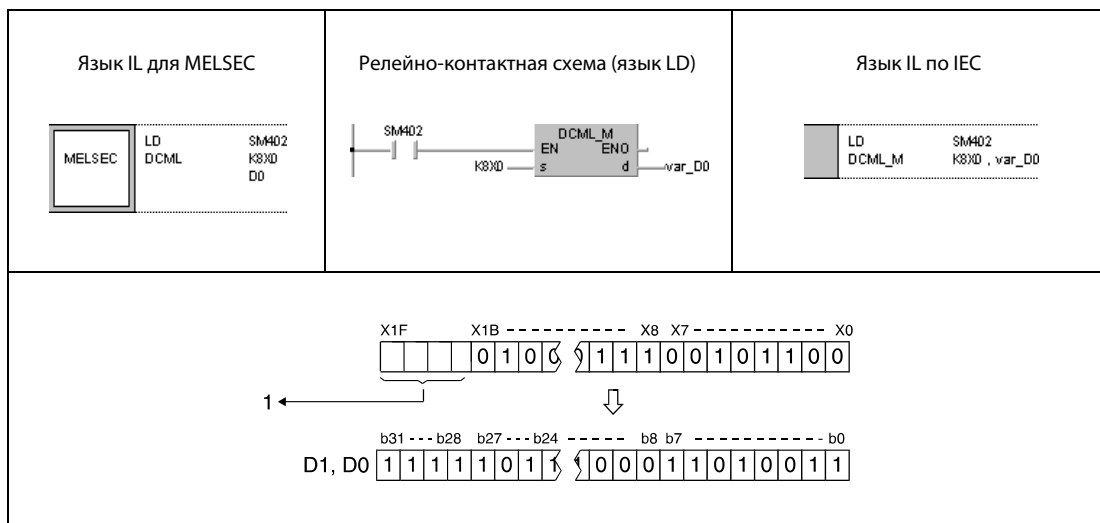
**Пример 3** CMLP

Следующая программа при положительном фронте сигнала X3 передает данные из D0 в инвертированном виде в D16.



**Пример 4** DCML

Следующая программа при включенном маркере SM402 передает данные X0...X1FF в инвертированном виде в D0 и D1.

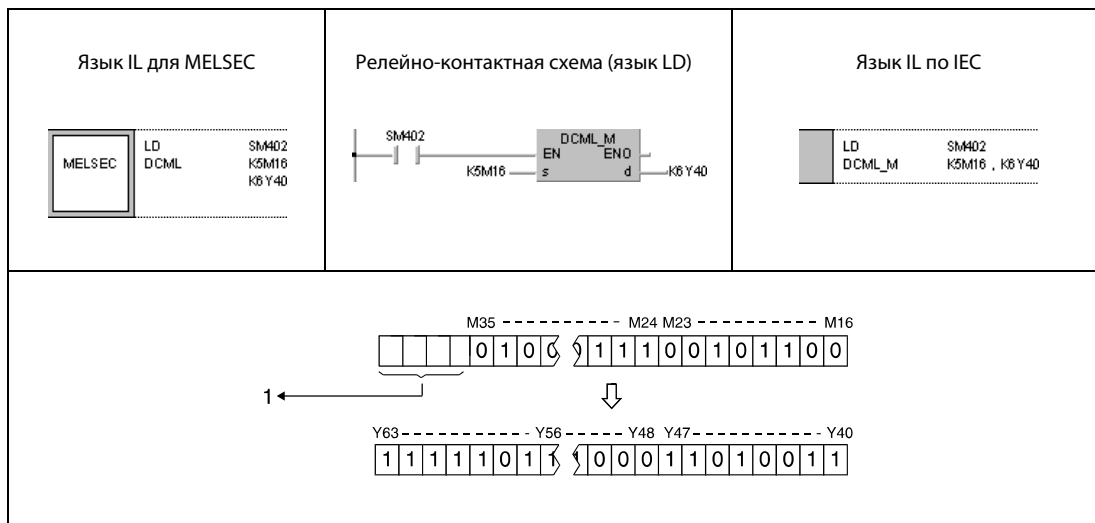


<sup>1</sup> Не указанные биты считываются в виде нулей.

Учитывайте, что в s должно быть меньше битов, чем в d.

**Пример 5** DCML

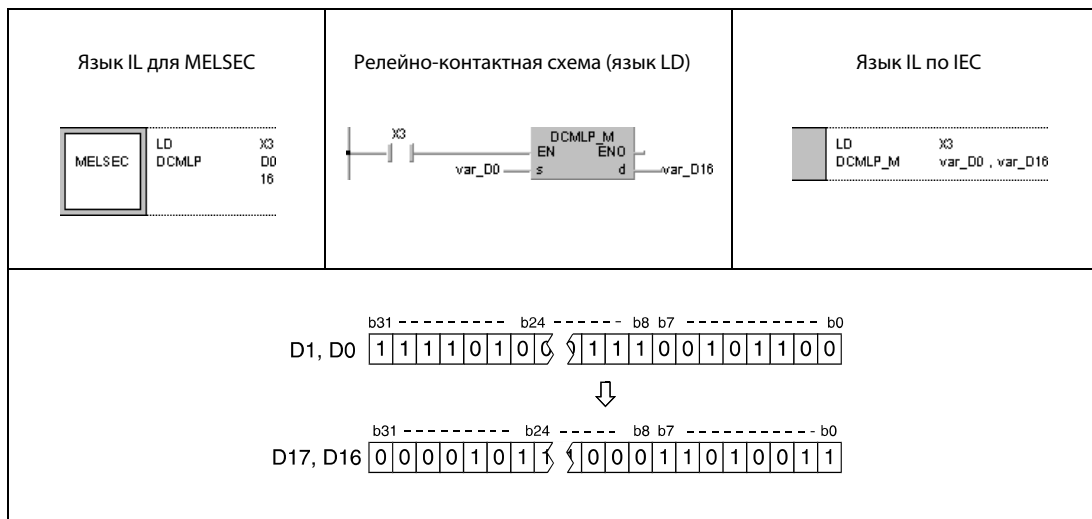
Следующая программа при включенном маркере SM402 передает данные M16...M35 в инвертированном виде в Y40 и Y57.



<sup>1</sup> Не указанные биты считываются в виде нулей.

**Пример 6** DCMLP

Следующая программа при положительном фронте сигнала X3 передает данные из D0 и D1 в инвертированном виде в D16 и D17.



**ПРИМЕЧАНИЯ** Количество битов в s должно быть меньше, чем количество битов в d.

Без объявления переменных в заголовке POU примеры программ 4 и 6 не работоспособны. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

6.4.5 BMOV, BMOVP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

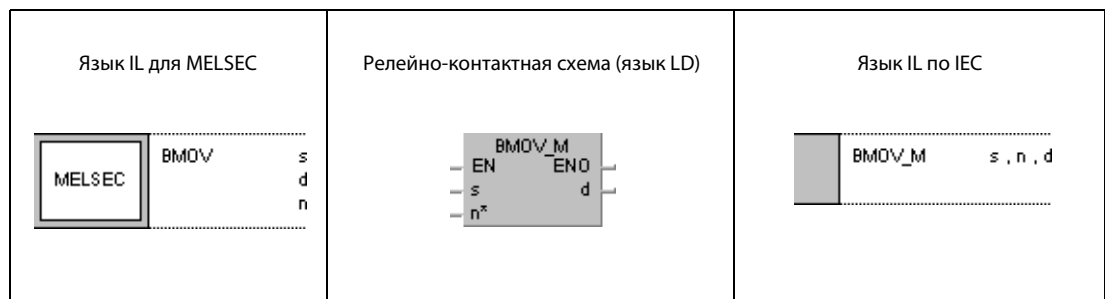
	Операнды															Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки			
	битовые					словные (16 бит)					Константы		Указатели		Уровень								
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z				V	K	H (16#)	P	I
s	●	●	●	●	●	●	●	●	●	●	●	●											
d		●	●	●	●	●	●	●	●	●													●
n																	●	●					

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

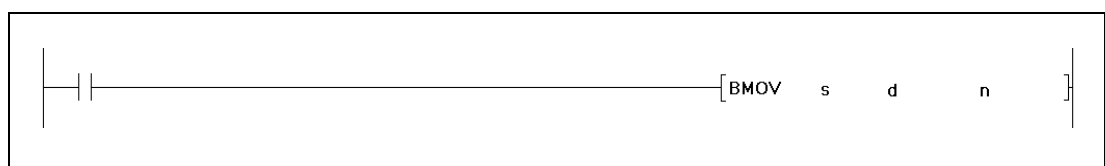
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.о.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	—	—	—	SM0	4
d	●	●	●	●	●	—	—	—			
n	●	●	●	●	●	●	●	—			

GX IEC Developer



GX Developer



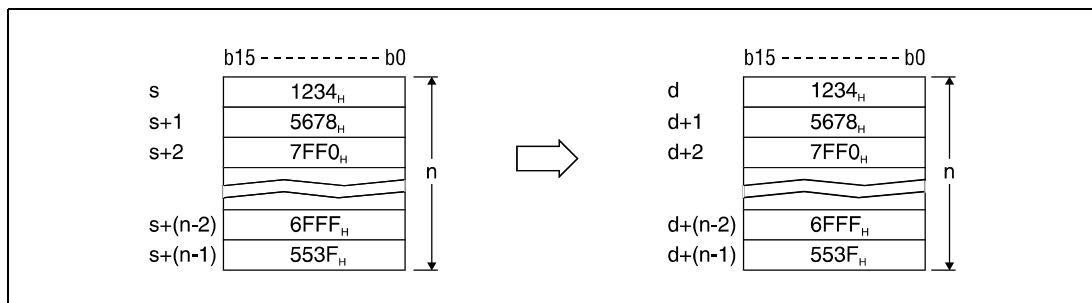
Переменные

Операнд	Значение	Тип данных
s	Операнд, в котором сохранены передаваемые данные.	BIN, 16 бит
d	Операнд, в который передаются данные.	
n	количество передаваемых блоков данных	

**Принцип действия**

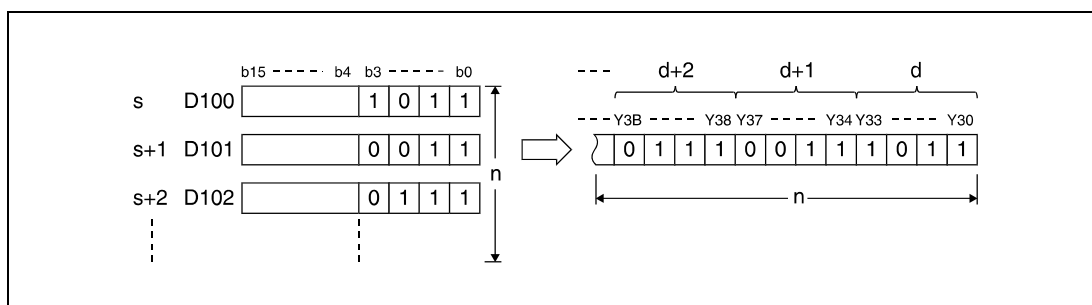
**Передача блоков двоичных данных**  
**BMOV Поблочная передача данных (16 бит)**

С помощью команды BMOV можно одновременно передать блок следующих друг за другом адресов. В s устанавливается первый передаваемый адрес. Значение в "n" указывает количество следующих друг за другом адресов. Данные передаются блоками по "n" адресов в целевой адрес, начиная с d.



Передача данных возможна даже в том случае, если источник и цель содержат одинаковые адреса. Передача в операнды с меньшим адресом начинается с s, а передача в операнды с более высоким адресом – с s + (n - 1).

Если s является словным операндом, а d – битовым, то в битовый операнд передаются разряды словного операнда, указанные путем обозначения битов. Например, если для d указан K1Y30, передаются 4 младших бита словного операнда, указанного в s.



Если адреса представляют собой битовые операнды, количество битов в d и s должно быть одинаковым.

**Источники ошибок**

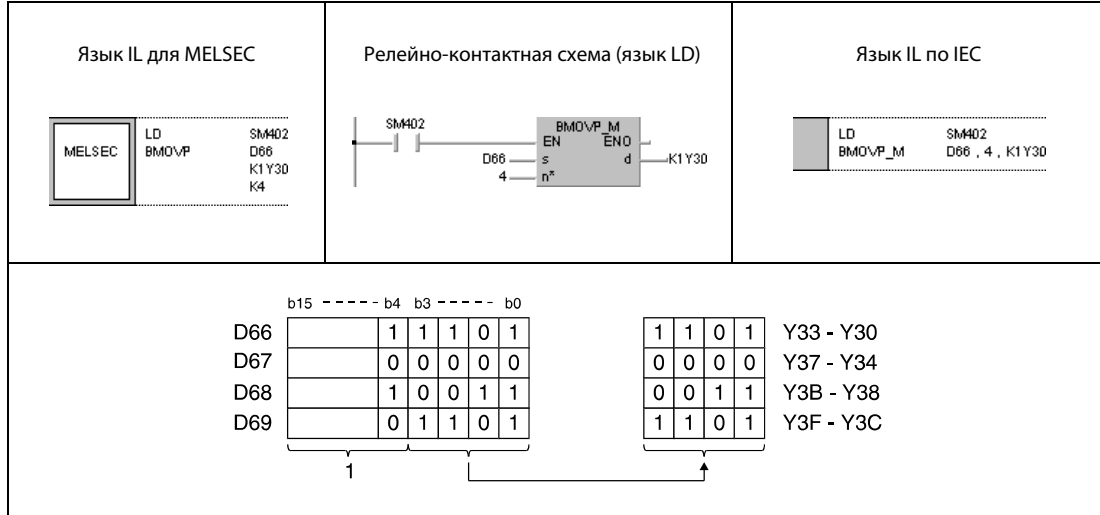
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в n количество блоков данных в s и d находится вне области операндов, предусмотренной для сохранения (серии "Q"/"System Q" = код ошибки 4101).

**Пример 1** ВMOVР

Следующая программа при положительном фронте сигнала SM402 передает данные младших 4 битов (от b0 до b3) регистров D66...D69 на выходы Y30...Y3F. Количество передаваемых блоков (4) указано константой K4.

На битовых схемах показана структура битов перед передачей и после передачи.

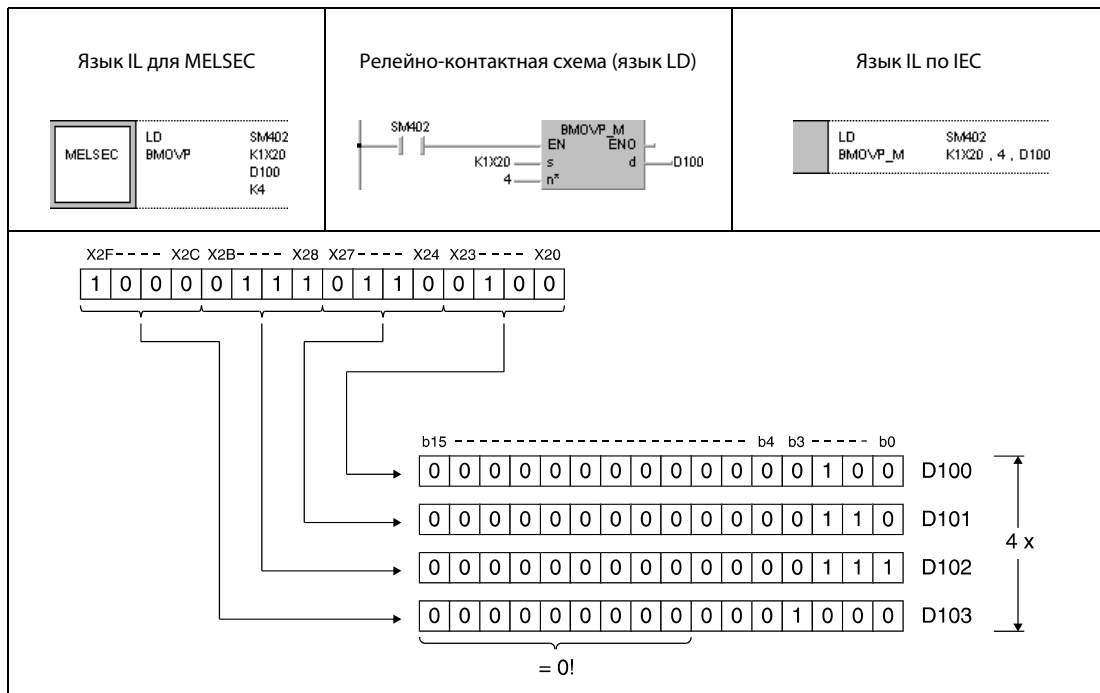


<sup>1</sup> Эти биты при операции игнорируются.

**Пример 2** ВMOVР

Следующая программа при положительном фронте сигнала SM402 передает данные X20...X2F в регистры D100...D103. Количество блоков данных (4) указано константой K4.

На битовых схемах показана структура битов перед передачей и после передачи.





### 6.4.6 FMOV, FMOVP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

	Операнды															Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки						
	битовые					словные (16 бит)					Константы		Указатели		Уровень											
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V	K	H (16#)	P	I	N
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	9 ↓ 1	●	M9012	M9010 M9011
d		●	●	●	●	●	●	●	●	●																
n																●	●									

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

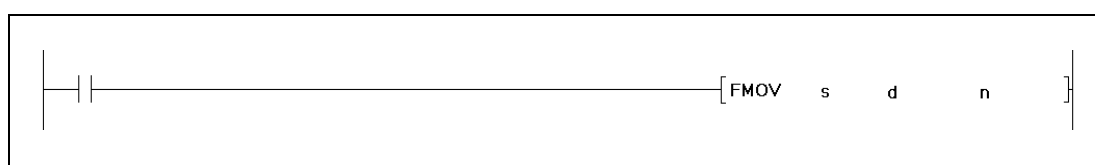
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.о.)		Регистр файлов	MELSECNET/10 непос. адр. J□\□□		Спец. модули U□\G□□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	●	—	SM0	4
d	●	●	●	●	●	—	—	—			
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer



GX Developer



Переменные

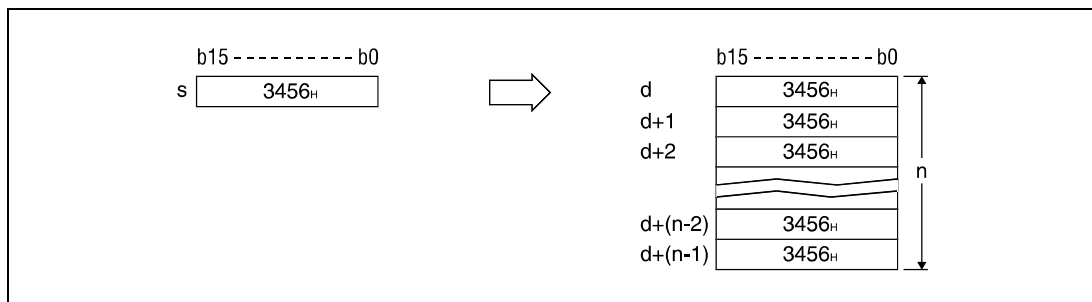
Операнд	Значение	Тип данных
s	Данные или операнд, в котором сохранены передаваемые данные.	BIN, 16 бит
d	Операнд, в который передаются данные.	
n	количество передач	

**Принцип действия**

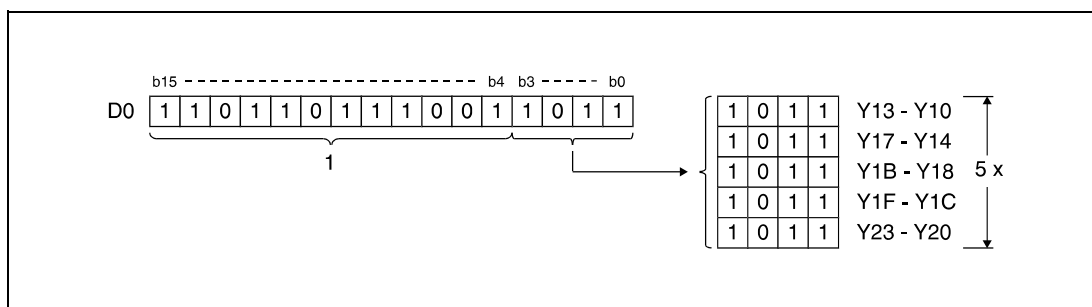
**Передача блока двоичных данных**

**FMOV Передача идентичного блока данных (16 бит)**

С помощью команды FMOV можно передать данные операнда s в область, которая начинается с операнда, установленного в d, и заканчивается операндом, установленным в d + (n - 1). В результате этого в блоке данных от d до d + (n - 1) каждый операнд устанавливается на значение операнда s.



Если s является словным операндом, а d – битовым, то в битовый операнд передаются разряды словного операнда, указанные путем обозначения битов.



<sup>1</sup> Эти биты при обработке не учитываются.

Если адреса представляют собой битовые операнды, количество битов в d и s должно быть одинаковым.

**Источники ошибок**

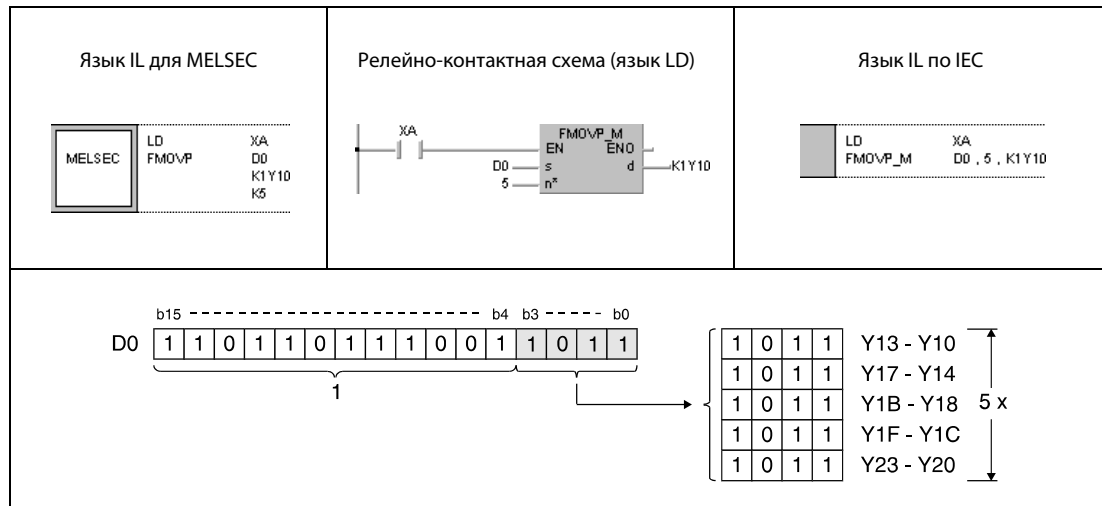
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в n количество блоков данных в s и d находится вне области операндов, предусмотренной для сохранения (серии "Q"/"System Q" = код ошибки 4101).

**Пример 1** FMOVP

Следующая программа при положительном фронте сигнала XA передает данные 4 младших битов (b0 – b3) D0 на выходы Y10...Y23. Количество блоков (5) указано константой K5.

На битовых схемах показана структура битов перед передачей и после передачи.

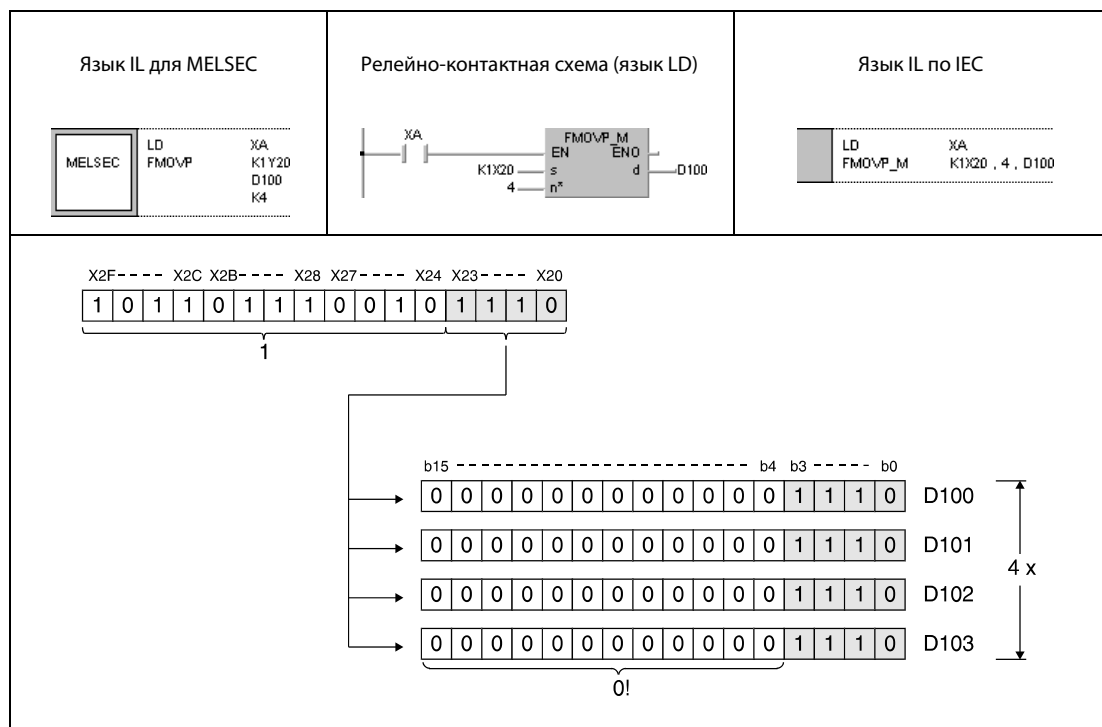


<sup>1</sup> Эти биты при обработке не учитываются.

**Пример 2** FMOVP

Следующая программа при положительном фронте сигнала XA передает данные X20...X23 в регистры данных D100...D103. Количество блоков (4) указано константой K4.

На битовых схемах показана структура битов перед передачей и после передачи.



<sup>1</sup> Эти биты при обработке не учитываются.

6.4.7 XCH, XCHP, DXCH, DXCHP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

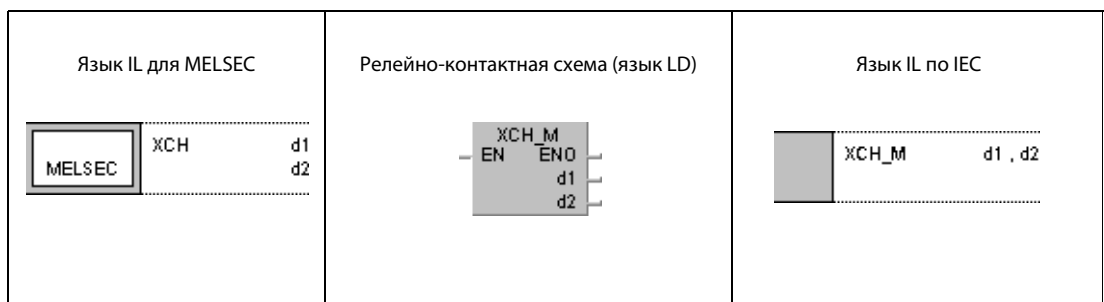
Операнды																Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки					
битовые				словные (16 бит)								Константы		Указатели							Уровень				
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V				K	H (16#)	P	I	N	M9012	M9011
XCH																									
d1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						K1 ↓ K4	5	●		●
d2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								1		
DXCH																									
d1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						K1 ↓ K8	7	●		●
d2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								1		●

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

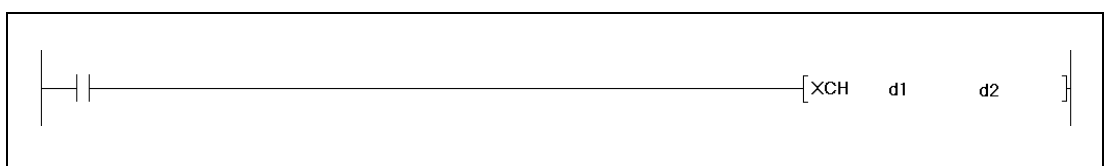
Операнды MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непос. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные							
d1	●	●	●	●	●	●	—	—	—	3	
d2	●	●	●	●	●	●	—	—	—		

GX IEC Developer



GX Developer



Переменные

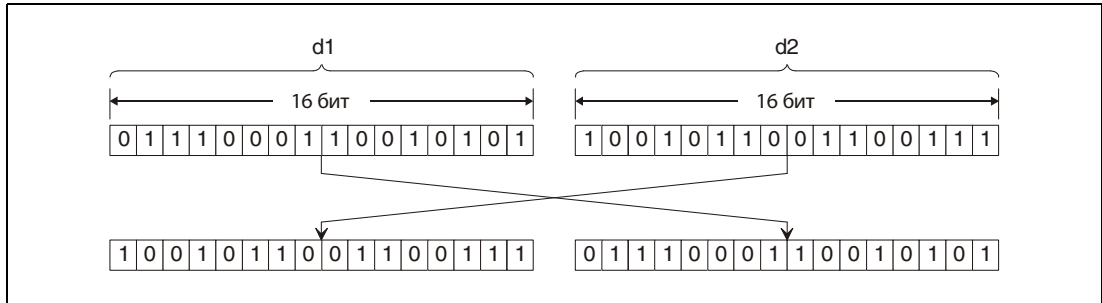
Операнд	Значение	Тип данных
d1	Операнды, в которых сохранены обмениваемые данные.	BIN, 16/32 бита
d2		

**Принцип действия**

**Обмен двоичными данными**

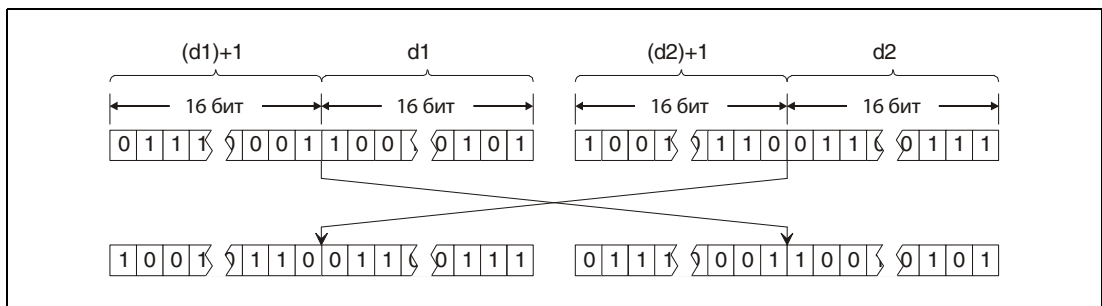
**XCH Обмен данными (16 бит)**

Команда XCH (от слова "eXChange" – "обменять") взаимно меняет местами 16-битные данные регистров d1 и d2.



**DXCH Обмен данными (32 бита)**

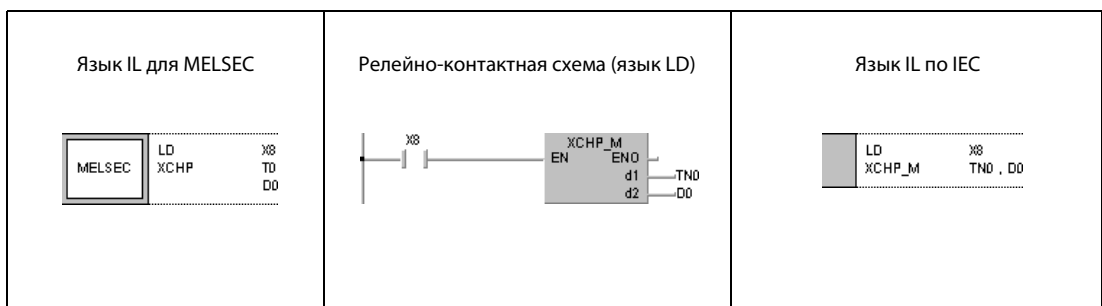
Команда DXCH взаимно меняет местами 32-битные данные регистров (d1) + 1, d1 и (d2) + 1, d2.



**Пример 1**

**XCHP**

Следующая программа при положительном фронте сигнала X8 заменяет содержимое регистра D0 фактическим значением таймера T0.

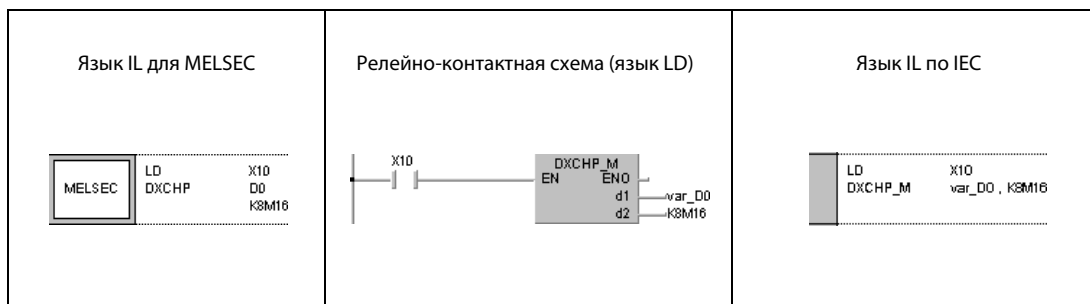


**Пример 2** XCHP

Следующая программа при положительном фронте сигнала X10 заменяет содержимое регистра D0 содержимым маркеров M16...M31.

**Пример 3** DXCHP

Следующая программа при положительном фронте сигнала X10 заменяет содержимое регистров D0 и D1 содержимым маркеров M16...M47.

**Пример 4** DXCHP

Следующая программа при положительном фронте M0 заменяет содержимое регистров D0 и D1 содержимым регистров D9 и D10.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU примеры программ 3 и 4 не работоспособны. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

**6.4.8 VXCH, VXCHP**

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

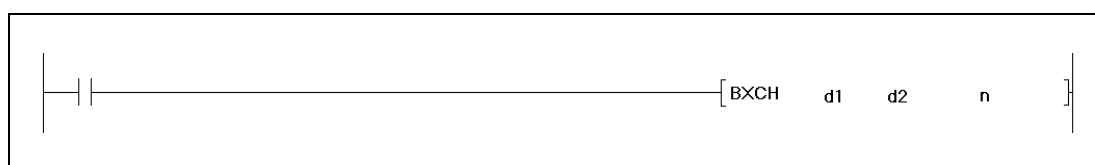
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
d1	—	●	●	—	—	—	—	—	—	SM0	4
d2	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

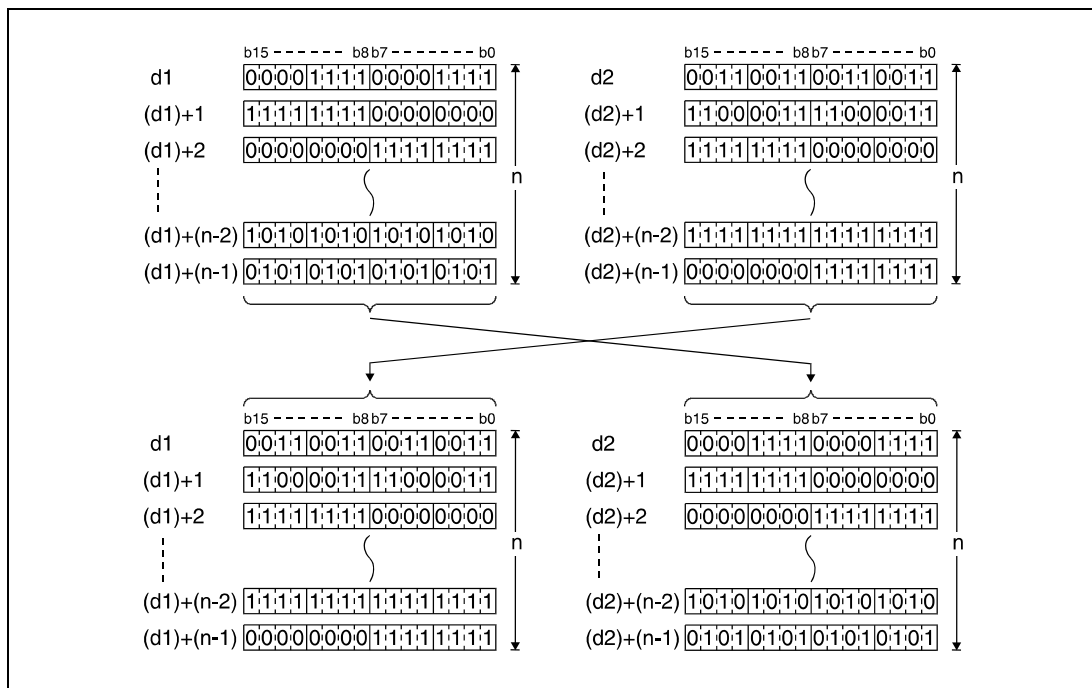
Операнд	Значение	Тип данных
d1	Операнды, в которых сохранены обмениваемые данные.	BIN, 16 бит
d2		
n	количество обмениваемых блоков данных	

**Принцип действия**

**Обмен блоками двоичных данных**

**VXCH Поблочный обмен данными (16 бит)**

Команда VXCH (от слова "eXCHange" – "обменять") взаимно меняет местами 16-битные данные блоков регистров d1 и d2.



**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в n количество блоков данных в d1 и d2 находится вне области операндов, предусмотренной для сохранения. (серии "Q"/"System Q" = код ошибки 4101).
- Области памяти d1 и d2 перекрываются (код ошибки 4101).

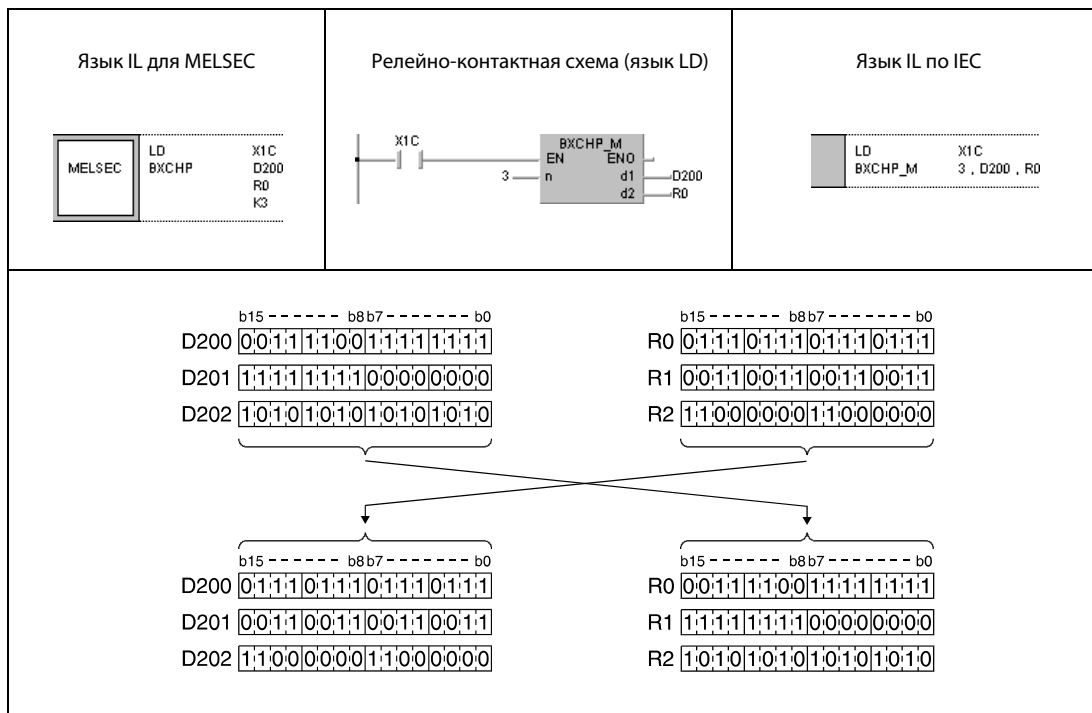


**Пример**

**ВХСНР**

Следующая программа при положительном фронте сигнала X1C меняет местами блоки данных, начиная с D200, с блоками данных, начиная с R0. Количество обмениваемых блоков двоичных данных (3) указано константой K3.

На битовых схемах показана структура битов перед передачей и после передачи.



### 6.4.9 SWAP, SWAPP

**Процессор**

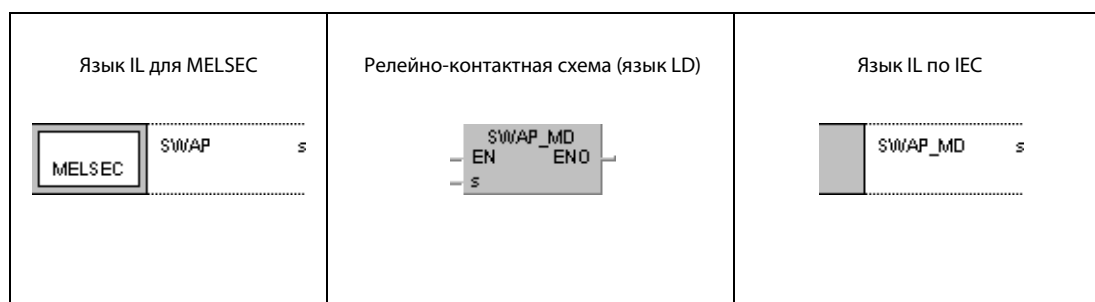
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	●

<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

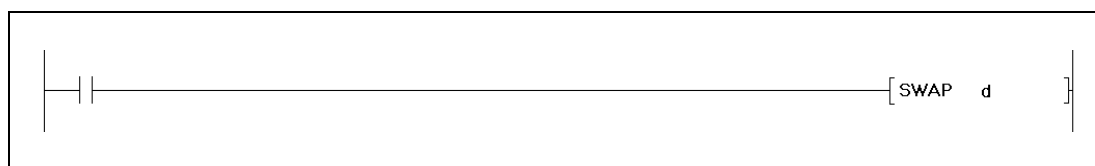
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	—	—	—	3

**GX IEC Developer**



**GX Developer**



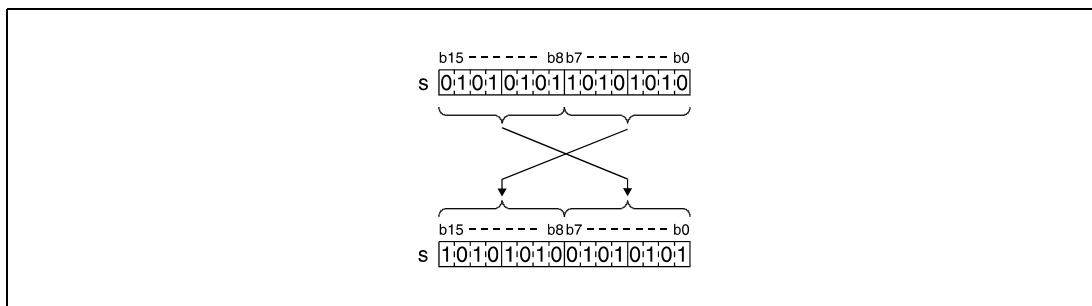
**Переменные**

Операнд	Значение	Тип данных
s	Операнд, в котором сохранены данные.	BIN, 16 бит

**Принцип действия**      **Перестановка байтов слова**

**SWAP    Перестановка старшего и младшего байта**

Команда SWAP меняет местами старшие 8 битов (старший байт) с младшими 8 битами (младшим байтом) слова, хранящегося в s.



**Пример**      **SWAPP**

Следующая программа при положительном фронте сигнала X10 меняет местами 8 старших битов с 8 младшими битами операнда R10.

Язык IL для MELSEC	Релейно-контактная схема (язык LD)	Язык IL по IEC
<p style="text-align: center;">             b15 ----- b8 b7 ----- b0              R10 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1              -----              b15 ----- b8 b7 ----- b0              R10 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0         </p>		



## 6.5 Команды разветвления программы

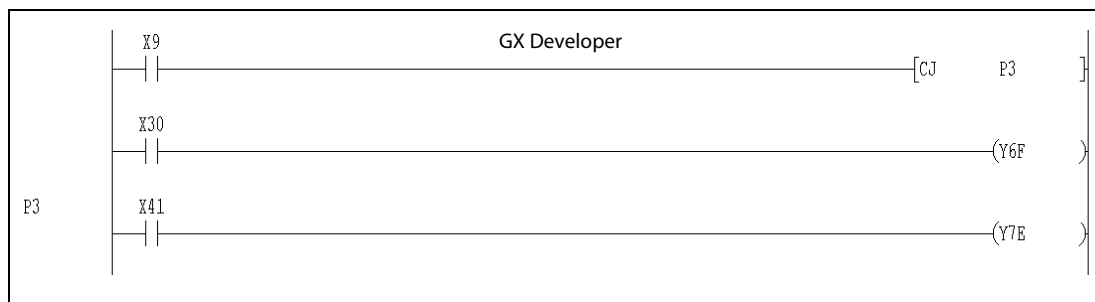
В командах разветвления программы указывается цель перехода.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC
Условный переход	CJ	CJ_M
Условный переход в следующем цикле	SCJ	SCJ_M
Команда перехода	JMP	JMP_M
Переход к концу программы	GOEND	GOEND_M

Цели перехода обозначаются указателем P (в MELSEC MEDOC и GX Developer) или меткой (в GX IEC Developer).

Подробные указания по программированию меток в GX IEC Developer имеются в руководстве GX IEC Developer.

MELSEC MEDOC	GX IEC Developer
LD X9	LD X9
CJ P3	JMPC Label_3
LD X30	LD X30
OUT Y6F	ST Y6F
P3	Label_3: LD X41
LD X41	ST Y7E
OUT Y7E	



### 6.5.1 CJ, SCJ, JMP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

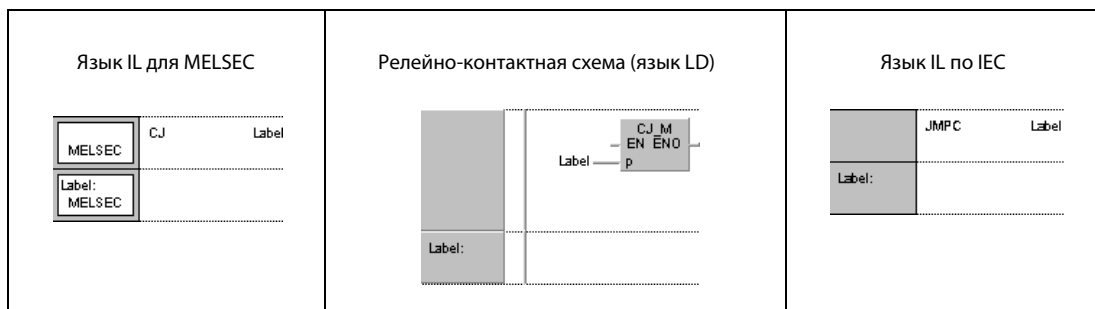
r	Операнды															Длина блока	Шагов	Индекс	Флаг переноса		Флаг ошибки						
	битовые							словные (16 бит)											Константы		Указатели		Уровень		M9012	M9010 M9011	
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z				V	K	H (16#)	P	I	N			
																			●				3	1	●		●

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

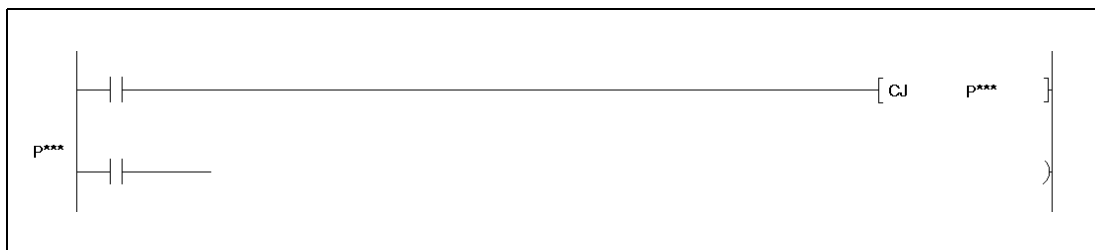
**Операнды MELSEC Q**

r	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
	—	—	—	—	—	—	—	—	●	SM0	2

**GX IEC Developer**



**GX Developer**



**Переменные**

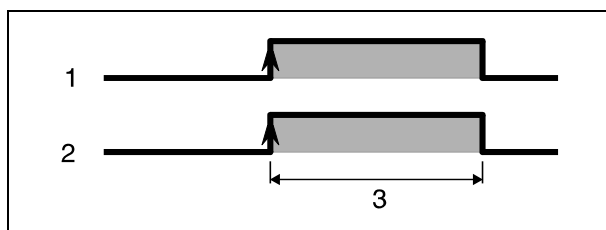
Операнд	Значение	Тип данных
r	Адрес перехода	Указатель/метка

**Принцип действия****Команды переходов**

Команда перехода состоит из слова CJ, SCJ или JMP (Conditional Jump = "условный скачок", JuMP = "скачок") и указателя (метки) P, обозначающего цель перехода. Адрес указателя (метки) можно использовать в следующих диапазонах: для процессоров серии "A" – от P (метка) 0 до P (метка) 255, при этом P (метка) 255 имеет значение команды END и не может служить в качестве адреса перехода. Для процессоров серий "Q" и "System Q" адрес указателя (метки) может находиться в диапазоне между P (метка) 0 и P (метка) 4095. Адрес перехода P(метка)xx можно размещать в любом месте программы.

**CJ Условный переход**

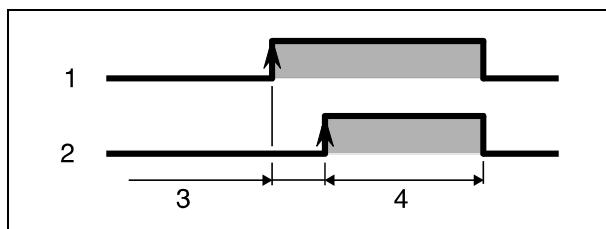
При выполненном входном условии команда CJ выполняет часть программы, находящуюся по указанному адресу перехода. Если входное условие не выполнено, выполняется следующий шаг программы.



- 1 входное условие
- 2 команда перехода
- 3 все циклы

**SCJ Условный переход при следующем прогоне**

При выполненном входном условии команда SCJ при следующем прогоне программы выполняет часть программы, находящуюся по указанному адресу перехода. Если входное условие не выполнено, выполняется следующий шаг программы.



- 1 входное условие
- 2 команда перехода
- 3 один цикл
- 4 каждый цикл

**JMP Команда перехода**

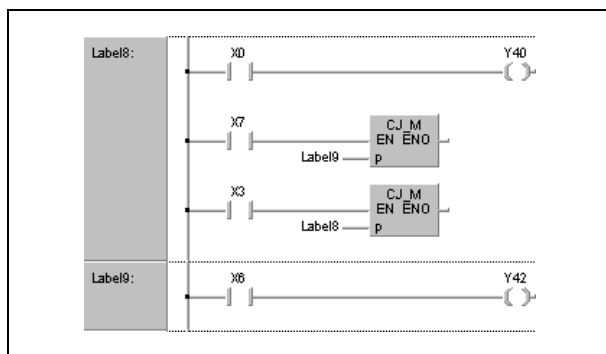
Команда JMP выполняет часть программы, находящуюся по указанному адресу перехода без входного условия (безусловный переход).

**ПРИМЕЧАНИЯ** Если в результате выполнения команды CJ, SCJ или JMP пропущен уже установленный таймер, таймер продолжает счет.

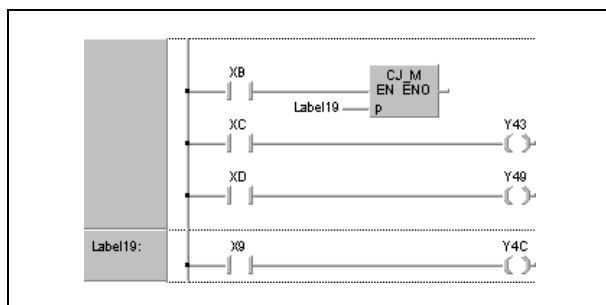
Если в результате выполнения одной из команд перехода пропущена команда OUT, состояние выхода остается без изменений.

При выполнении команды перехода время цикла программы укорачивается в соответствии с количеством пропущенных шагов программы. (см. раздел А.2.1 „Перечень значений времени обработки (серии QnA и System Q)“ в приложении)

С помощью команд CJ, SCJ и JMP возможен и "возврат" к более низкому адресу перехода. Однако при составлении программы необходимо обращать внимание на то, чтобы выход из замкнутого цикла происходил прежде, чем истечет время сторожевого таймера (в примере выход из цикла происходит при включении X7).



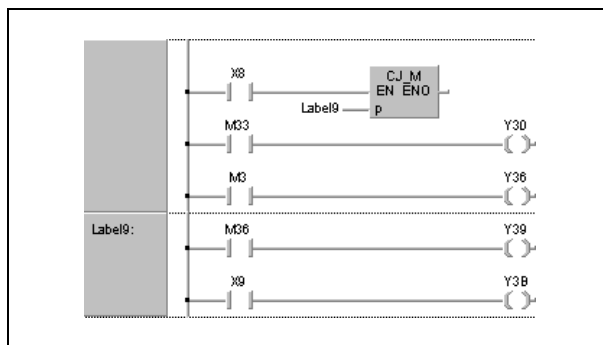
Операнд, пропущенный в результате выполнения команды перехода, не изменяет своего состояния. Это наглядно показано на нижеследующем примере.



После включения X8 происходит переход по адресу Label19. Состояния выходов Y43 и Y49 остаются без изменений даже при включении или выключении XC или XD.



Адрес перехода Label9 занимает один шаг в программе.



С помощью команды CJ, SCJ или JMP возможен переход только по адресу, находящемуся в этой же программе.

Если адрес перехода находится в пропускаемой области (т. е. области, в отношении которой действует операция пропуска части программы), то программа продолжается с ближайшего возможного адреса, следующего за адресом перехода.

**Источники ошибок**

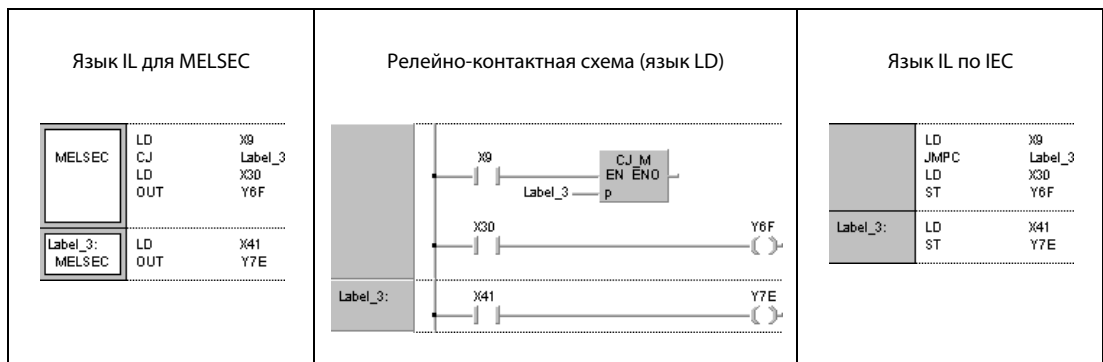
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Адрес указателя встречается в программе дважды и к нему обращается команда перехода (серии "Q"/"System Q" = код ошибки 4210).
- Указанный в команде перехода адрес перехода в программе не определен (адрес перехода или указатель отсутствует) (серии "Q"/"System Q" = код ошибки 4210).
- Цель перехода находится за командой END (серии "Q"/"System Q" = код ошибки 4210).
- Цель перехода находится внутри цикла FOR/NEXT.
- Цель перехода находится в подпрограмме.

**Пример 1**

CJ

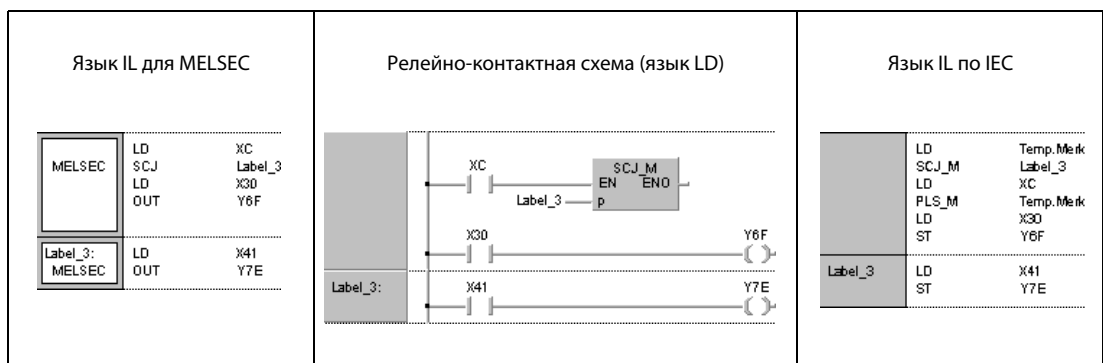
В следующей программе при включении входа X9 выполняется переход к адресу перехода Label\_3.



**Пример 2**

SCJ

В этом примере переход к адресу Label\_3 происходит в следующем цикле после включения XC.



### 6.5.2 GOEND

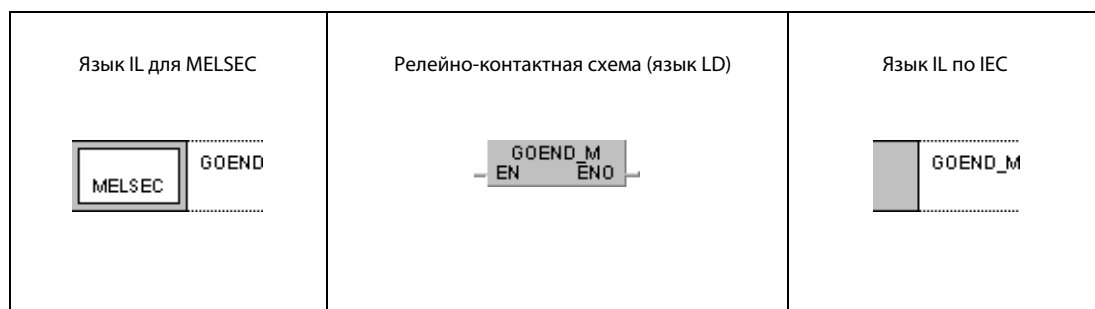
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

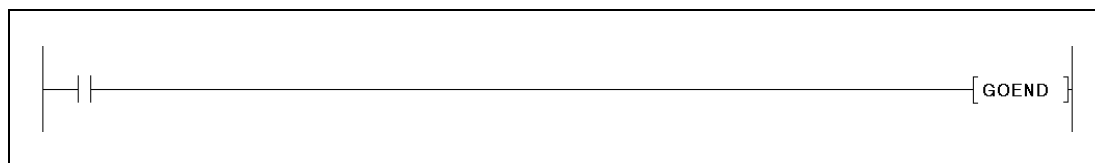
**Операнды MELSEC Q**

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные							
—	—	—	—	—	—	—	—	—	—	SMO	1

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
—	—	—

**Принцип действия**

**GOEND Переход к концу программы**

Адресом перехода команды GOEND является команда FEND или END программы.

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Команда GOEND выполнена после выполнения команды CALL или ECALL и перед выполнением команды RET (код ошибки 4211).
- Команда GOEND выполнена после выполнения команды FOR и перед выполнением команды NEXT (код ошибки 4200).
- Команда GOEND выполнена в подпрограмме прерывания перед выполнением команды IRET (код ошибки 4221).
- Команда GOEND выполнена между командами CHK CIR и CHK END (код ошибки 4230).
- Команда GOEND выполнена между командами IX и IX END (код ошибки 4231).

**Пример**

GOEND

В следующей программе при отрицательном значении в D0 происходит переход к команде END.



## 6.6 Команды для вызова программ прерывания

Эти команды позволяют вызывать подпрограммы прерывания. При этом прерывания можно активировать или деактивировать по отдельности или на основе битовых схем. В следующей таблице дан обзор команд.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC
Прерывание деактивировано	DI	DI_M
Прерывание разрешено	EI	EI_M
Битовая схема условий выполнения программ прерывания	IMASK	IMASK_M
Конец программы прерывания	IRET	IRET_M

6.6.1 DI, EI, IMASK

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
● <sup>1</sup>	● <sup>1</sup>	●	●	●	●

<sup>1</sup> В процессоре AnN или AnS команда EI/DI выполнима только в том случае, если не установлен специальный маркер M9053.

Операнды MELSEC A

Операнды														Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011				
битовые						словные (16 бит)						Константы							Указатели		Уровень	
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1						Z	V	K	H (16#)

<sup>1</sup> только для команд EI и DI

Операнды MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные							
s	—	● <sup>1</sup>	—	—	—	—	—	—	—	—	2 ● <sup>1</sup>
—	—	—	—	—	—	—	—	—	—	—	1 ● <sup>2</sup>

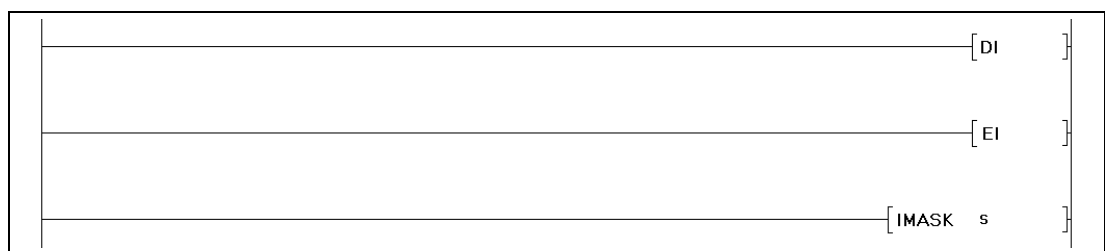
<sup>1</sup> только для команды IMASK

<sup>2</sup> только для команд EI и DI

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
s	Битовая схема, в которой указаны условия выполнения прерывания, или начальный адрес, начиная с которого хранится битовая схема.	BIN, 16 бит

**Принцип действия****Команды прерывания**

Программа прерывания представляет собой вставленную часть программы (обозначенную адресом прерывания Ixx), которая может вызываться внешним сигналом прерывания. Программа прерывания выполняется в зависимости от команд EI/DI. В случае применения процессора AnN функция команд EI/DI зависит от состояния специального маркера M9053. Только если этот маркер не установлен, эти команды служат в качестве условий для выполнения программы прерывания. Если маркер M9053 установлен, то эти команды используются для обновления канала связи (см. раздел "Команды обновления данных" этого руководства).

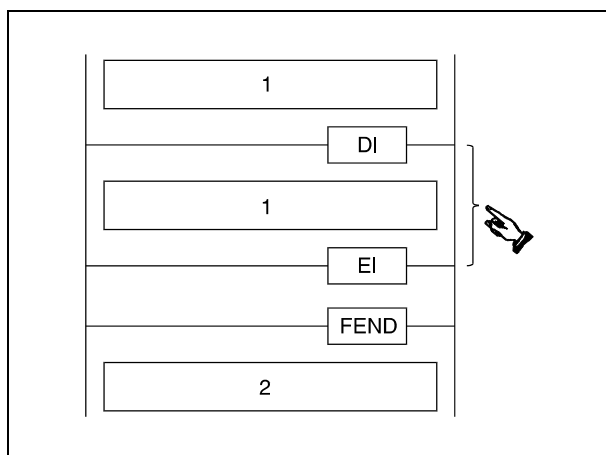
**DI      Дезактивация прерываний**

Команда DI предотвращает выполнение программы прерывания до тех пор, пока среди команд программы не встретится команда EI. После включения или сброса центрального процессора он находится в состоянии DI.

**EI      Активация прерываний**

Команда EI позволяет вызывать программы прерывания путем указания адреса прерывания Ixx или выполнения команды IMASK.

Если прерывание происходит между командами DI и EI, то это прерывание выполняется лишь после отработки фрагмента программы, расположенного в промежутке между этими командами DI и EI. Это пояснено на следующей иллюстрации.



<sup>1</sup> основная программа

<sup>2</sup> программа прерывания

**ПРИМЕЧАНИЕ**

*В среде GX IEC Developer команда FEND вставляется автоматически. Событие Ixx должно быть сопоставлено какой-либо задаче.*

### IMASK Битовые схемы условий выполнения программ прерывания (только серии "Q" и "System Q")

В битовой схеме, указанной в  $s$ , каждому биту сопоставлен определенный адрес прерывания. От состояния соответствующего бита зависит, возможно ли выполнение соответствующего прерывания. Если бит имеет значение 0, сопоставленная программа прерывания выполняться не может. Если бит имеет состояние 1, программа прерываний выполняется.

#### Процессор QnA

На следующей иллюстрации показано соответствие между битами в данных от  $s$  до  $s + 2$  и адресами прерывания.

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
$s$	115	114	113	112	111	110	19	18	17	16	15	14	13	12	11	10
$s+1$	131	130	129	128	127	126	125	124	123	122	121	120	119	118	117	116
$s+2$	147	146	145	144	143	142	141	140	139	138	137	136	135	134	133	132

После включения центрального процессора или его сброса с помощью выключателя с ключом "RUN-STOP" биты  $b0 - b31$  (адреса прерывания  $I0 - I31$ ) установлены на 1, т. е. эти программы прерывания могут выполняться. Биты  $b32 - b47$  (адреса прерывания  $I32 - I47$ ) имеют состояние 0, т. е. соответствующие программы прерывания выполняться не могут.

Битовые схемы, указанные в данных от  $s$  до  $s + 2$ , сохраняются в регистрах диагностики с SD715 по SD717.

#### Однопроцессорный модуль ЦП серии "Q"

Биты в данных от  $s$  до  $s + 7$  сопоставлены адресам прерывания следующим образом:

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
$s$	115	114	113	112	111	110	19	18	17	16	15	14	13	12	11	10
$s+1$	131	130	129	128	127	126	125	124	123	122	121	120	119	118	117	116
$s+2$	147	146	145	144	143	142	141	140	139	138	137	136	135	134	133	132
$s+3$	163	162	161	160	159	158	157	156	155	154	153	152	151	150	149	148
$s+4$	179	178	177	176	175	174	173	172	171	170	169	168	167	166	165	164
$s+5$	195	194	193	192	191	190	189	188	187	186	185	184	183	182	181	180
$s+6$	1111	1110	1109	1108	1107	1106	1105	1104	1103	1102	1101	1100	199	198	197	196
$s+7$	1127	1126	1125	1124	1123	1122	1121	1120	1119	1118	1117	1116	1115	1114	1113	1112

После включения напряжения питания центрального процессора или его сброса с помощью выключателя "RUN/STOP" выполнение программ прерывания с  $I0$  по  $I31$  разрешено.

Битовые схемы, указанные в данных от  $s$  до  $s + 2$ , сохраняются в регистрах диагностики с SD715 по SD717. Битовые схемы от  $s + 3$  до  $s + 7$  сохраняются в регистрах диагностики с SD781 по SD785. Хотя используются регистры диагностики в двух отдельных диапазонах номеров (от SD715 до SD717 и от SD781 до SD785), битовые схемы имеют непрерывную нумерацию от  $s$  до  $s + 7$ .



**Многопроцессорный модуль ЦП серии "Q"**

На следующей иллюстрации показано соответствие между битами данных  $s...s + 15$  и адресами прерывания.

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
s	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
s + 1	I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
s + 2	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
s + 3	I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
s + 4	I79	I78	I77	I76	I75	I74	I73	I72	I71	I70	I69	I68	I67	I66	I65	I64
s + 5	I95	I94	I93	I92	I91	I90	I89	I88	I87	I86	I85	I84	I83	I82	I81	I80
s + 6	I111	I110	I109	I108	I107	I106	I105	I104	I103	I102	I101	I100	I99	I98	I97	I96
s + 7	I127	I126	I125	I124	I123	I122	I121	I120	I119	I118	I117	I116	I115	I114	I113	I112
s + 8	I143	I142	I141	I140	I139	I138	I137	I136	I135	I134	I133	I132	I131	I130	I129	I128
s + 9	I159	I158	I157	I156	I155	I154	I153	I152	I151	I150	I149	I148	I147	I146	I145	I144
s + 10	I175	I174	I173	I172	I171	I170	I169	I168	I167	I166	I165	I164	I163	I162	I161	I160
s + 11	I191	I190	I189	I188	I187	I186	I185	I184	I183	I182	I181	I180	I179	I178	I177	I176
s + 12	I207	I206	I205	I204	I203	I202	I201	I200	I199	I198	I197	I196	I195	I194	I193	I192
s + 13	I223	I222	I221	I220	I219	I218	I217	I216	I215	I214	I213	I212	I211	I210	I209	I208
s + 14	I239	I238	I237	I236	I235	I234	I233	I232	I231	I230	I229	I228	I227	I226	I225	I224
s + 15	I255	I254	I253	I252	I251	I250	I249	I248	I247	I246	I245	I244	I243	I242	I241	I240

После включения напряжения питания центрального процессора или после его сброса (с помощью выключателя "RUN/STOP") разрешено выполнение программ прерывания с I0 по I31 и с I48 по I255. Программы прерывания с I32 по I47 заблокированы и их выполнение невозможно.

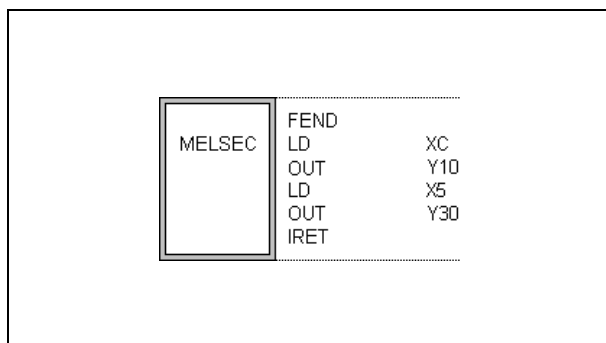
Битовые схемы, указанные в данных от  $s$  до  $s + 2$ , сохраняются в регистрах диагностики с SD715 по SD717. Битовые схемы, указанные от  $s + 3$  до  $s + 15$ , сохраняются в регистрах диагностики с SD781 по SD793.

Хотя регистры диагностики распределены по разнесенным диапазонам номеров с SD715 по SD717 и с SD781 по SD793, битовые схемы имеют непрерывную нумерацию от  $s$  до  $s + 15$ .

**ПРИМЕЧАНИЯ** Если в программу прерывания потребовалось вставить счетчик, то для этого должен использоваться специальный счетчик для программы прерывания. Центральные процессоры типов A3H, A3M, AnA, AnAS и AnU не имеют счетчиков, которые можно использовать в программе прерывания.

Адрес прерывания (указатель прерывания) для обозначения программы прерывания занимает один шаг программы.

Не вкладывайте программы прерывания одну в другую, т. е. не вызывайте из программы прерывания другую программу прерывания.



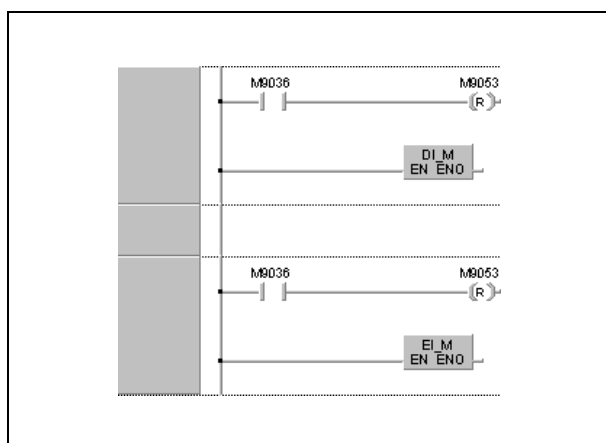
При написании программ в среде GX Developer или в редакторе Melsec среды GX IEC Developer команды FEND и IRET программируются пользователем.

При программировании в редакторе IEC прерывание сопоставляется какой-либо задаче, после чего компилятор среды GX IEC Developer автоматически устанавливает команды FEND и IRET (см. пример программирования).

Если во время выполнения программы прерывания внутри процессора устанавливается состояние DI, то одновременно не могут выполняться никакие другие программы прерывания. Вызов другой программы прерывания возможен лишь после очередной установки команды EI.

Если команда EI или DI находится в пределах команды MC, то эта команда EI или DI выполняется без учета команды MC.

В процессорах AnN или AnS команда EI/DI выполняется только в том случае, если не установлен специальный маркер M9053. Если этот специальный маркер установлен, команда EI/DI интерпретируется как условие обновления канала связи. Чтобы команду EI/DI можно было использовать в качестве условия для вызова программ прерывания в процессорах AnN или AnS, перед командой EI/DI необходимо сбрасывать специальный маркер M9053.



**Пример** EI, DI, IMASK (GX IEC Developer)

В следующем примере программы после включения входа X0 становится возможным выполнение программы прерывания. При выключенном X0 выполнение программы прерывания деактивировано.

На нижней иллюстрации показаны задачи, программируемые в редакторе IEC и вызывающие программы прерывания I1 и I2.

Interrupt\_1 (I1) и Interrupt\_2 (I2) являются программами прерывания. Команду IRET программировать не требуется, так как ее автоматически вставляет компилятор среды GX IEC Developer.

<p style="text-align: center;">Язык IL для MELSEC</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%; padding: 2px;">MELSEC</td> <td style="padding: 2px;">LD X0</td> <td style="padding: 2px;">CJ Label_1</td> <td style="padding: 2px;">DI</td> </tr> <tr> <td style="padding: 2px;">Label_1: MELSEC</td> <td style="padding: 2px;">LD X0</td> <td style="padding: 2px;">CJ Label_2</td> <td style="padding: 2px;">LD X0</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;">MOVP H6</td> <td style="padding: 2px;">D10</td> <td style="padding: 2px;">MOVP H0</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;">MOVP H0</td> <td style="padding: 2px;">D11</td> <td style="padding: 2px;">MOVP H0</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;">MOVP H0</td> <td style="padding: 2px;">D12</td> <td style="padding: 2px;">IMASK EI</td> </tr> <tr> <td style="padding: 2px;">Label_2: MELSEC</td> <td style="padding: 2px;">LD M0</td> <td style="padding: 2px;">OUT Y20</td> <td></td> </tr> </table>	MELSEC	LD X0	CJ Label_1	DI	Label_1: MELSEC	LD X0	CJ Label_2	LD X0		MOVP H6	D10	MOVP H0		MOVP H0	D11	MOVP H0		MOVP H0	D12	IMASK EI	Label_2: MELSEC	LD M0	OUT Y20		<p style="text-align: center;">Релейно-контактная схема (язык LD)</p>
MELSEC	LD X0	CJ Label_1	DI																						
Label_1: MELSEC	LD X0	CJ Label_2	LD X0																						
	MOVP H6	D10	MOVP H0																						
	MOVP H0	D11	MOVP H0																						
	MOVP H0	D12	IMASK EI																						
Label_2: MELSEC	LD M0	OUT Y20																							
<p style="text-align: center;">Язык IL по IEC</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;"></td> <td style="padding: 2px;">LD X0</td> <td style="padding: 2px;">JMPC Label_1</td> <td style="padding: 2px;">DI_M</td> </tr> <tr> <td style="padding: 2px;">Label_1:</td> <td style="padding: 2px;">LD X0</td> <td style="padding: 2px;">JMPC Label_2</td> <td style="padding: 2px;">LD X0</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;">MOVP_M H6</td> <td style="padding: 2px;">var_D10[10]</td> <td style="padding: 2px;">MOVP_M H0</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;">MOVP_M H0</td> <td style="padding: 2px;">var_D10[11]</td> <td style="padding: 2px;">MOVP_M H0</td> </tr> <tr> <td style="padding: 2px;"></td> <td style="padding: 2px;">MOVP_M H0</td> <td style="padding: 2px;">var_D10[12]</td> <td style="padding: 2px;">IMASK_M EI_M</td> </tr> <tr> <td style="padding: 2px;">Label_2:</td> <td style="padding: 2px;">LD M0</td> <td style="padding: 2px;">ST Y20</td> <td></td> </tr> </table>		LD X0	JMPC Label_1	DI_M	Label_1:	LD X0	JMPC Label_2	LD X0		MOVP_M H6	var_D10[10]	MOVP_M H0		MOVP_M H0	var_D10[11]	MOVP_M H0		MOVP_M H0	var_D10[12]	IMASK_M EI_M	Label_2:	LD M0	ST Y20		
	LD X0	JMPC Label_1	DI_M																						
Label_1:	LD X0	JMPC Label_2	LD X0																						
	MOVP_M H6	var_D10[10]	MOVP_M H0																						
	MOVP_M H0	var_D10[11]	MOVP_M H0																						
	MOVP_M H0	var_D10[12]	IMASK_M EI_M																						
Label_2:	LD M0	ST Y20																							
<div style="border: 1px solid black; padding: 5px;"> <p><b>Task Information</b></p> <p>Task Attributes</p> <p>Event: I1</p> <p>Interval: 0</p> <p>Priority: 31</p> <p>Name: Interrupt_1</p> <p>Size: 87 Bytes</p> <p>Type: TASK <input checked="" type="checkbox"/> Timer/ Output Control</p> <p>Last Change: 06.10.1997 14:59:36</p> <p>Security Level: 0 1 2 3 4 5 6 7</p> <p><input checked="" type="checkbox"/> Allow Read Access for lower Levels</p> </div>	<div style="border: 1px solid black; padding: 5px;"> <p><b>Task Information</b></p> <p>Task Attributes</p> <p>Event: I2</p> <p>Interval: 0</p> <p>Priority: 31</p> <p>Name: Interrupt_2</p> <p>Size: 87 Bytes</p> <p>Type: TASK <input checked="" type="checkbox"/> Timer/ Output Control</p> <p>Last Change: 06.10.1997 15:00:05</p> <p>Security Level: 0 1 2 3 4 5 6 7</p> <p><input checked="" type="checkbox"/> Allow Read Access for lower Levels</p> </div>																								

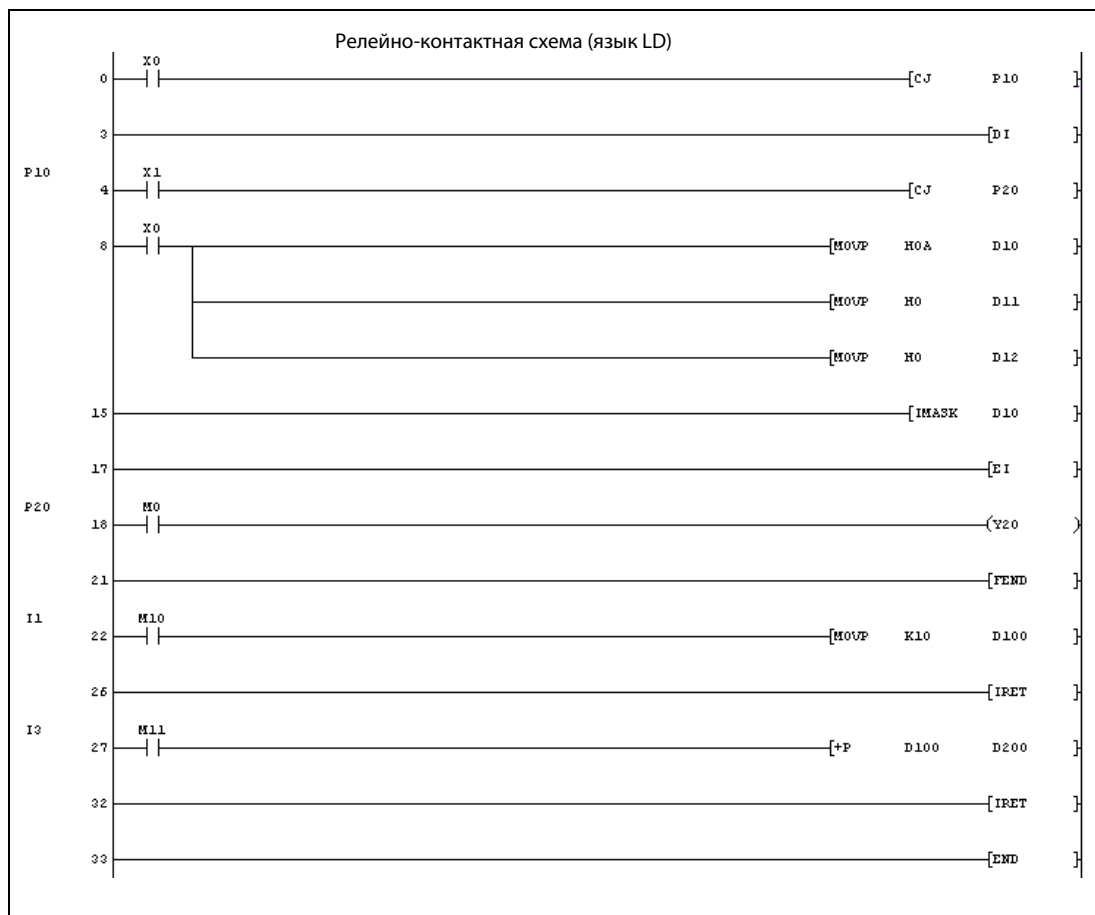
**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию можно найти в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

**Пример**

EI, DI, IMASK в случае процессора "Q" (GX Developer)

В следующем примере программы после включения входа X0 становится возможным выполнение программы прерывания. При выключенном X0 выполнение программы прерывания деактивировано.

I1 и I3 являются программами прерывания.



Список инструкций (язык IL)

0	LD	X0	
1	CJ	P10	
3	DI		
4	P10		
5	LD	X1	
6	CJ	P20	
8	LD	X0	
9	MOVP	H0A	D10
11	MOVP	H0	D11
13	MOVP	H0	D12
15	IMASK	D10	
17	EI		
18	P20		
19	LD	M0	
20	OUT	Y20	
21	FEND		
22	I1		
23	LD	M10	
24	MOVP	K10	D100
26	IRET		
27	I3		
28	LD	M11	
29	+P	D100	D200
32	IRET		
33	END		

### 6.6.2 IRET

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

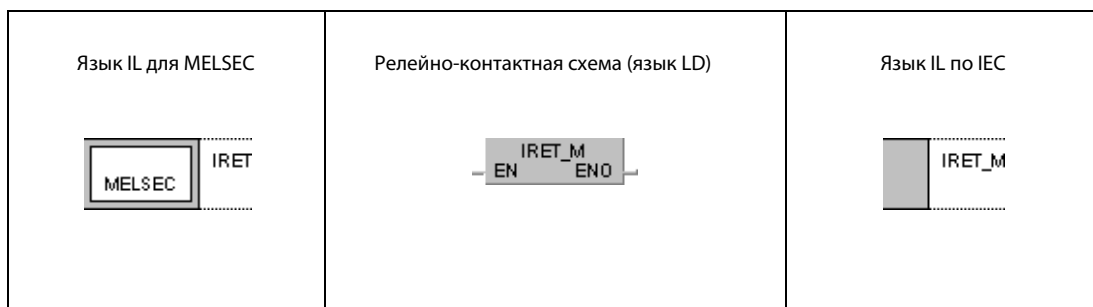
Операнды MELSEC A

Операнды															Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки					
битовые					словные (16 бит)					Константы		Указатели		Уровень										
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z				V	K	H (16#)	P	I	N	
																					1		M9012	M9010 M9011

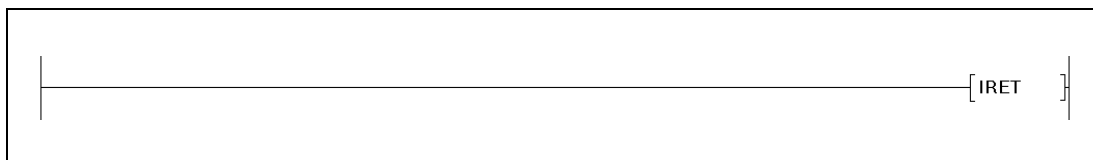
Операнды MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□V□		Спец. модули U□VG□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные							
—	—	—	—	—	—	—	—	—	—	—	1

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
—	—	—

**ПРИМЕЧАНИЕ** В редакторах по стандарту IEC команда IRET вырабатывается автоматически.

**Принцип действия****Возврат из программы прерывания в основную программу****IRET      Конец программы прерывания**

Конец программы прерывания обозначается командой IRET.

Обработка счетчиков во время выполнения программы прерывания продолжается.

Возврат к основной программе происходит после выполнения команды IRET.

В процессорах A3H, A3M, AnA, AnAS и AnU счетчики программ прерывания не обрабатываются.

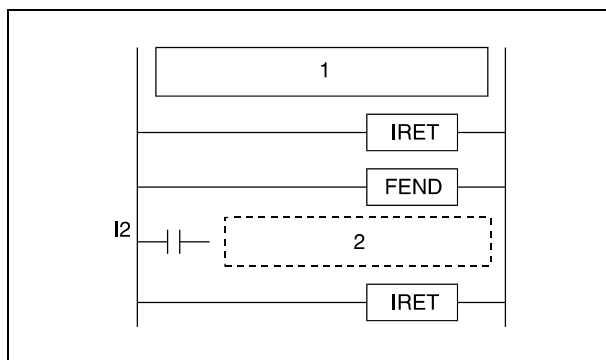
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Отсутствует целевой адрес для вызова прерывания (серии "Q"/"System Q" = код ошибки 4220).
- Если команда IRET расположена перед программой прерывания, процессор прекращает обработку в этом месте (серии "Q"/"System Q" = код ошибки 4223).
- После вызова программы прерывания и перед выполнением команды IRET установлена команда END, FEND, GOEND или STOP (серии "Q"/"System Q" = код ошибки 4221).

**ПРИМЕЧАНИЕ**

*В следующем примере показано ошибочное программирование!*



1 основная программа

2 программа прерывания

**Пример**

Применение команды IRET в программе продемонстрировано в примере для команд EI, DI, IMASK.

## 6.7 Команды обновления данных

Команды обновления данных позволяют обновлять данные интерфейсов ввода-вывода или данные процессов передачи. В следующей таблице дан обзор команд обновления данных.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC
Частичное обновление ввода-вывода (серия "Q")	RFS	RFS_M
	RFSP	RFSP_M
Частичное обновление ввода-вывода (серия "A")	SEG	SEG_M
Команда обновления сетевых и интерфейсных данных	COM	COM_M
Условия выполнения для команды обновления сетевых и интерфейсных данных	EI	EI_M
	DI	DI_M

6.7.1 RFS, RFSP

Процессор

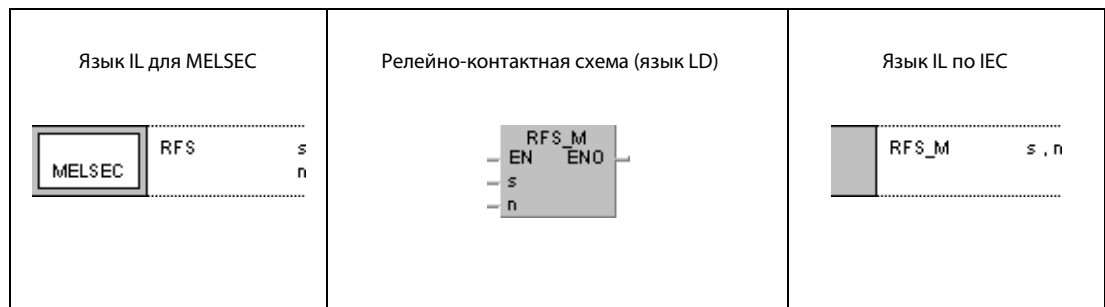
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Операнды MELSEC Q

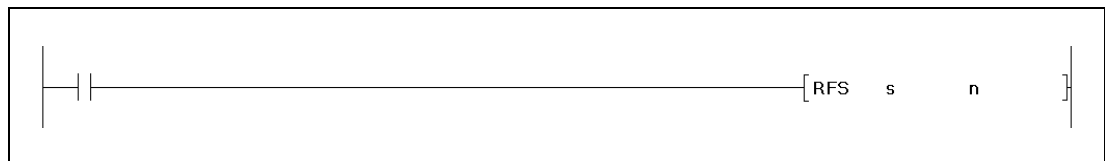
	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s	● <sup>1</sup>	—	—	—	—	—	—	—	—	SM0	3
n	●	●	●	●	●	●	●	●	—		

<sup>1</sup> только X и Y

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
s	Начальный адрес ввода-вывода, в отношении которого применяется команда обновления.	бит
n	количество обновляемых входных/выходных битов	BIN, 16 бит



**Принцип действия**

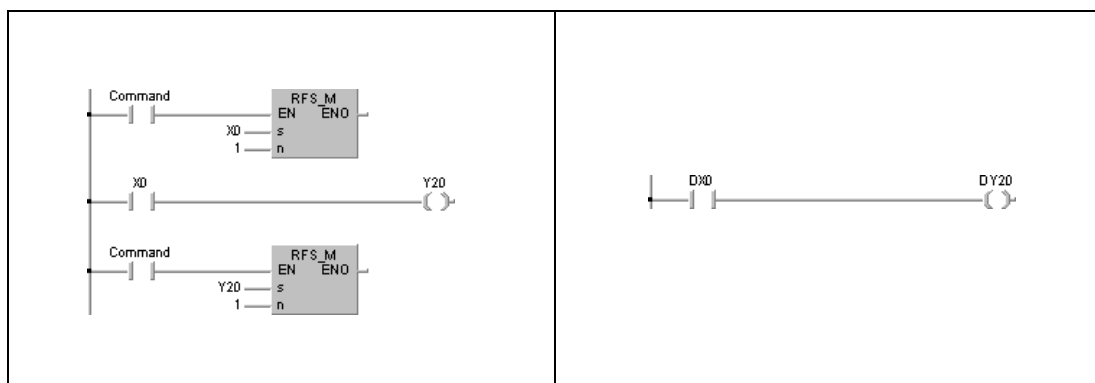
**Частичное обновление ввода-вывода (серии "Q" и "System Q")**

**RFS Команда обновления**

Команда RFS обновляет данные входов и выходов выбранного диапазона на один программный цикл и применяется для считывания данных внешнего источника или вывода данных на выходной модуль.

Считывание данных внешнего источника или вывод данных на внешний источник происходит в виде пакета после выполнения команды END. Поэтому во время прогона программы вывод импульсных сигналов не возможен. При частичном обновлении с помощью команды SEG заданные адреса входов (X) и выходов (Y) для ввода-вывода обновляются отдельно. Таким способом можно выводить и импульсные сигналы.

При использовании непосредственно адресуемых входов и выходов (DX/DY) входы (X) и выходы (Y) обновляются бит за битом.



В левом примере команда RFS обновляет вход X0 и выход Y20.

В правом примере та же функция выполняется в отношении входа DX и выхода DY без применения команды обновления.

**Источники ошибок**

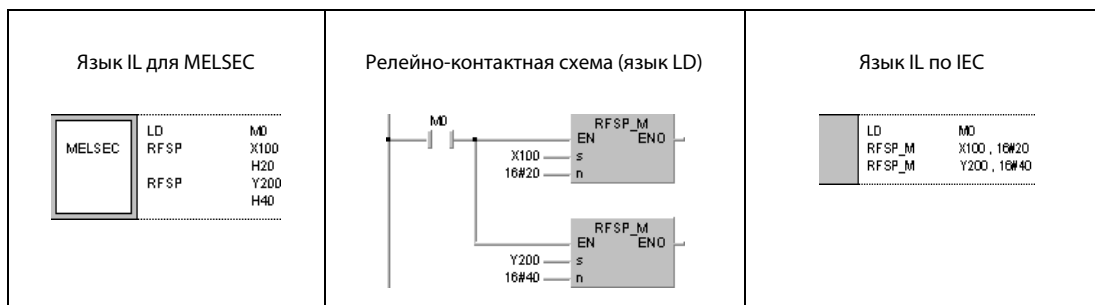
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в n количество обновляемых битов находится вне диапазона адресов ввода-вывода.

**Пример 1**

**RFSP**

Следующая программа при положительном фронте M0 обновляет входы X100...X11F и выходы Y200...Y23F.



### 6.7.2 SEG

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
● <sup>1</sup>	● <sup>1</sup>	● <sup>1</sup>	● <sup>1</sup>		

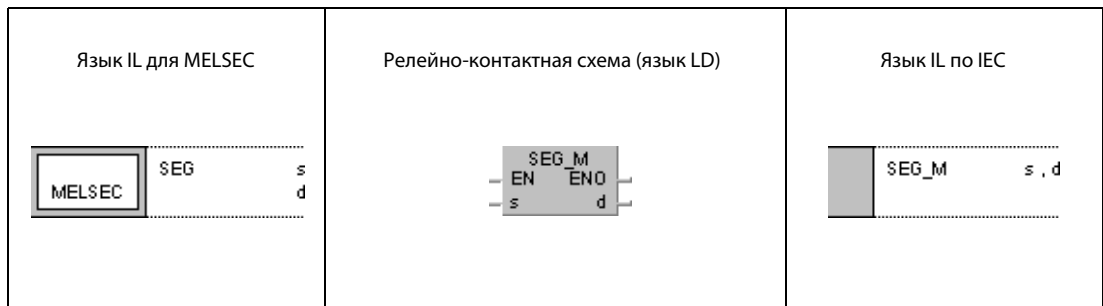
<sup>1</sup> Частичное обновление возможно только при установленном маркере M9052.

**Операнды MELSEC A**

		Операнды																	Длина блока К1 ↓ K8	Шагов 9 ● <sup>1</sup>	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011		
		битовые							словные (16 бит)							Константы		Указатели						Уровень	
		X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K				H (16#)	P	I	N
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							
d		●	●	●	●	●	●	●	●	●	●	●	●		●										

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

**GX IEC Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Начальный адрес входа/выхода, в отношении которого применяется команда обновления.	бит
d	Количество обновляемых входных/выходных битов.	BIN, 16 бит

**Принцип действия****Частичное обновление ввода-вывода (серия "А")****Общие сведения**

Частичное обновление возможно только при установленном специальном маркере M9052. Если этот специальный маркер не установлен, команда SEG имеет функцию 7-сегментной декодировки.

**SEG Частичное обновление**

Команда SEG позволяет обновлять определенный диапазон адресов ввода-вывода при выполненном входном условии.

При частичном обновлении заданные адреса ввода-вывода обновляются только на один программный цикл. В это время входные сигналы продолжают приниматься, а выходные сигналы продолжают передаваться в выходные модули.

При обработке входов и выходов в режиме обновления, в результате частичного обновления состояние входов (X) и выходов (Y) изменяется на один программный цикл.

Во время обычного обновления канала связи входные и выходные сигналы обрабатываются пакетно после выполнения команды END. Поэтому во время прогона программы вывод импульсных сигналов не возможен. При частичном обновлении с помощью команды SEG заданные адреса входов (X) и выходов (Y) для ввода-вывода обновляются отдельно. Таким способом можно выводить и импульсные сигналы.

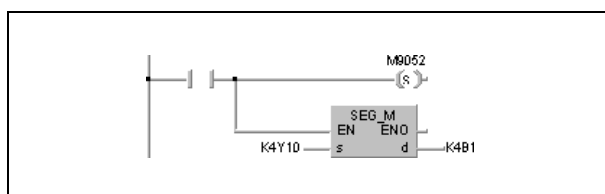
**ПРИМЕЧАНИЕ**

*В программе для процессора A2C во время коммуникации с модулями ввода-вывода в ходе выполнения программы импульсные сигналы не могут выдаваться даже в том случае, если выполняется частичное обновление выходов (Y) с помощью команды SEG. Более подробную информацию на эту тему можно найти в описании аппаратуры серии A2C.*

**Условия выполнения****Структура программы**

Программирование следует начать с установки специального маркера M9052. Исходные данные s, следующие за командой SEG, указывают начальный адрес частичного обновления (только для входов X и выходов Y). Кроме исходных данных устанавливается количество входов и выходов блоками по 8 входов или выходов.

На рисунке показана схема программы для команды SEG.

**Начальный адрес**

Начальный адрес – это первый адрес череды входных или выходных операндов (например, X0, X10, Y20 и т. п.).

Если указан адрес между Yn0 и Yn7 (Xn0 и Xn7), то обновление происходит, начиная с адреса Yn0 (Xn0). Если указан адрес между Yn8 и YnF (Xn8 и XnF), то обновление происходит, начиная с адреса Yn8.

Число адресов

Число обновляемых адресов может быть в диапазоне от 8 до 2048. Блоки по 8 адресов устанавливаются следующим образом:

V1 = 8 адресов

V2 = 16 адресов

...

VA = 80 адресов

VB = 88 адресов

...

V10 = 128 адресов

...

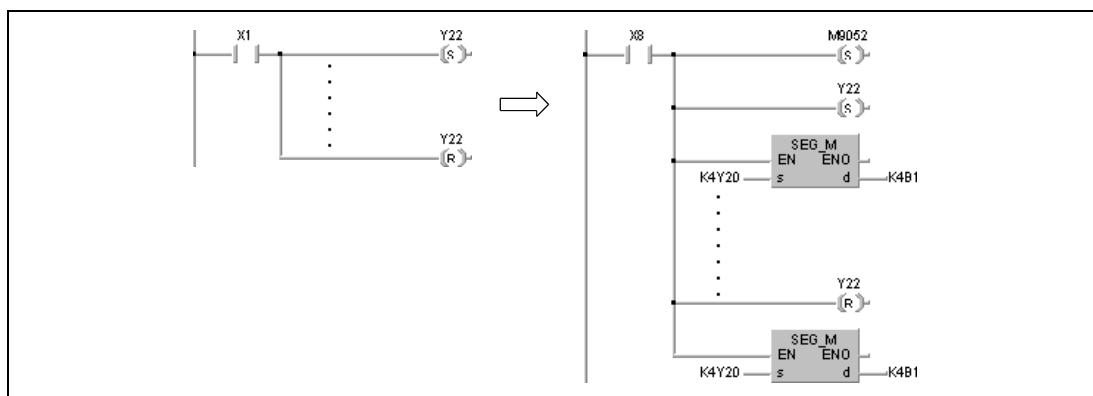
VFF = 2048 адресов

При указании V0 обновляются все адреса контроллера, начиная с заданного начального адреса.

Частичное обновление продолжается даже в том случае, если команда SEG выполняется в непосредственном режиме процессора. Однако в этом случае состояния входов и выходов не изменяются.

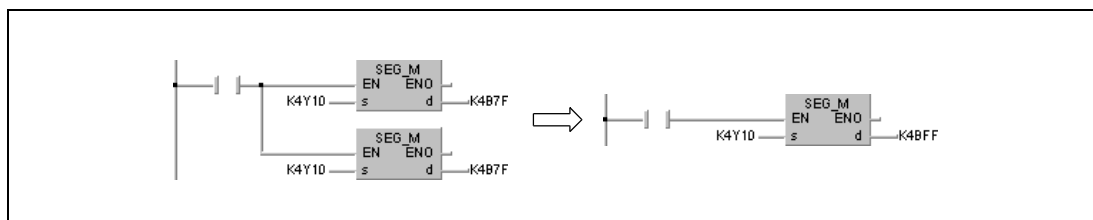
При переходе из непосредственного режима управления входами и выходами в режим внутреннего отображения входов и выходов (режим обновления) импульсный вывод с помощью команд SET и RST необходимо изменить следующим образом.

**ПРИМЕЧАНИЕ** Процессор A2C нижеприведенную программу выполнить не может.



**ПРИМЕЧАНИЕ** Если в программе для процессора AnA или AnU команда SEG обращается сразу ко всем 2048 адресам, то обновление входов и выходов может работать неправильно. Поэтому их следует распределить на 2 группы по 1024 адреса.

В нижеприведенном примере показана разбивка программы для обновления 2048 адресов в программном цикле процессора AnA или AnU.



**Пример**

SEG

В следующей программе обновляются входы X10...X1F.



### 6.7.3 COM

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

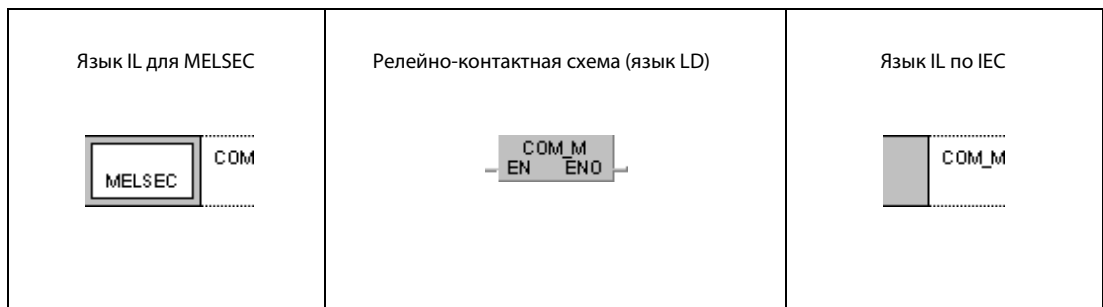
**Операнды MELSEC A**

Операнды																Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки					
битовые						словные (16 бит)						Константы		Указатели							Уровень				
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V				K	H (16#)	P	I	N	M9012	M9010 M9011
																							3		

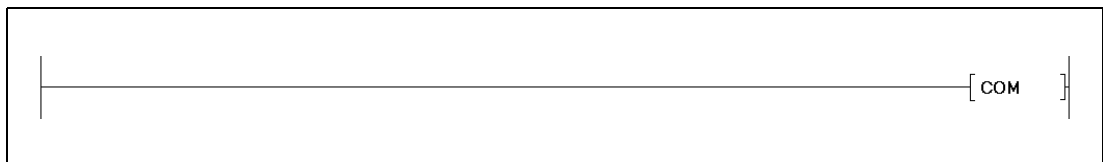
**Операнды MELSEC Q**

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непоср. адр. J□V□		Спец. модули U□V□	Индексные регистры Z <sub>n</sub>	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные							
—	—	—	—	—	—	—	—	—	—	—	1

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
—	—	—

**Принцип действия****Обновление канала связи****COM Команда обновления сетевых и интерфейсных данных**

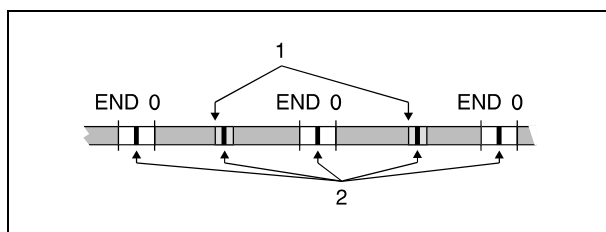
Характер выполнения команды COM в процессорах серий QnA и "System Q" зависит от состояния маркера диагностики SM775:

- Если SM775 не установлен (0), происходит обновление сетевых и интерфейсных данных (обновление канала связи) и обработка всех данных (обработка команды END).
- Если SM775 установлен (1), выполняется только обработка всех данных (обработка команды END).

Нижеследующие описания действительны для серий QnA и "System Q" в случае не установленного маркера SM775 (0), а также для серии "A":

Команда COM применяется, например, для ускорения обмена данными с удаленной станцией ввода-вывода. Если длительность цикла программы мастер-станции превышает длительность цикла локальной станции, то команда COM позволяет правильно обрабатывать принятые входные и выходные данные.

При выполнении команды COM центральный процессор временно прерывает основную программу, а затем обрабатывает все данные (обработка команды END) и обновляет сетевые и интерфейсные данные (обновление канала связи).



<sup>1</sup> команда COM

<sup>2</sup> обработка всех данных и обновление канала связи

Команду COM можно применять в программе сколь угодно часто. Однако учитывайте, что время цикла программы увеличивается на время обработки всех данных и обновления канала связи.

**ПРИМЕЧАНИЯ**

*Во время обработки всех данных происходят следующие процессы:*

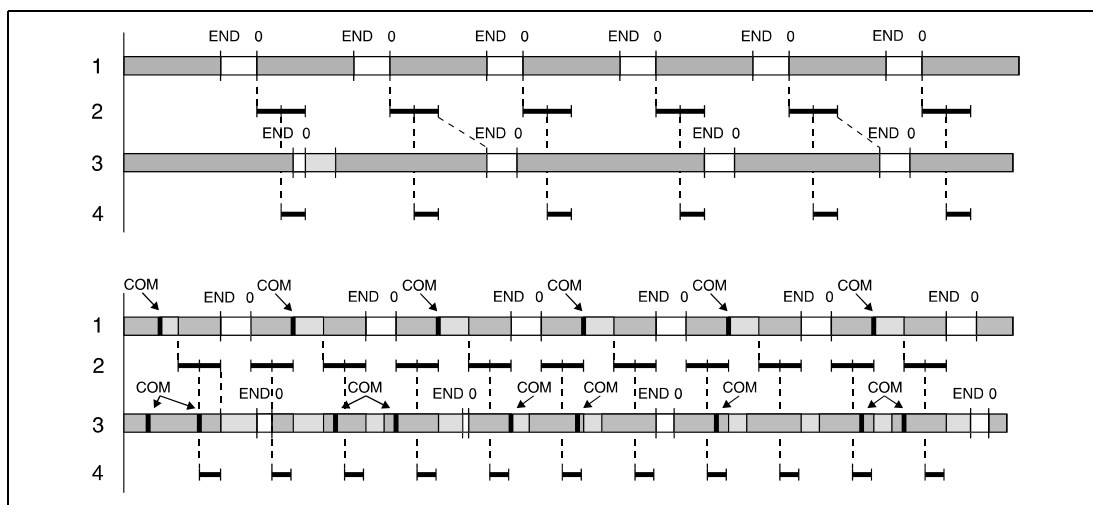
- коммуникация между контроллером и коммуникационными модулями,
- контроль прочих станций,
- считывание буферной памяти прочих специальных модулей через модуль Computer-Link.

*При обновлении сетевых и интерфейсных данных происходят следующие процессы:*

- обновление CC-Link,
- автоматическое обновление специальных модулей,
- обновление MELSECNET/10 и MELSECNET/H,
- автоматическое обновление общей области памяти в многопроцессорной системе (только в процессорах Q02, Q02H, Q06H, Q12H, Q12PH, Q25H и Q25PH, начиная с версии "B" программного обеспечения).

**Условия выполнения:**

В верхней части иллюстрации показаны временные диаграммы передачи данных без команды COM. В нижней части изображены временные диаграммы при использовании команды COM.



- 1 программа мастер-станции
- 2 передача данных
- 3 программа локальной станции
- 4 удаленная станция ввода-вывода, обновление входов и выходов

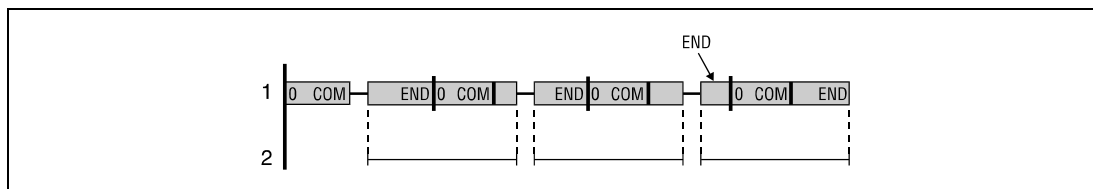
Команда COM в программе мастер-станции позволяет ускорить передачу данных в сети, так как возрастает количество сеансов коммуникации с удаленными станциями ввода-вывода.

Если длительность цикла программы локальной станции больше, чем в мастер-станции, то в верхнем примере правильный прием данных не возможен. В этом случае надежный обмен данными обеспечивается путем применения команды COM в программе локальной станции.

Если в программе локальной станции запрограммирована команда COM, то обновление канала связи происходит всякий раз, когда локальная станция получает от мастер-станции команду в промежутке между следующими командами:

- шаг 0 и команда COM,
- две команды COM,
- команда COM и команда END.

Если длительность цикла сети больше, чем длительность цикла программы мастер-станции, то ускорить обмен данными не возможно (даже вставив команду COM в программу главной станции).



- 1 программа мастер-станции
- 2 время цикла подчиненной станции (Slave)



### 6.7.4 EI, DI

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
● <sup>1</sup>	● <sup>1</sup>				

<sup>1</sup> Обновление канала связи возможно только при установленном маркере M9053.

**Операнды MELSEC A**

Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011			
битовые						словные (16 бит)						Константы		Указатели							Уровень		
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V							K	H (16#)
																					1		

**GX IEC Developer**

Язык IL для MELSEC	Релейно-контактная схема (язык LD)	Язык IL по IEC																
<div style="border: 1px solid black; padding: 5px; width: fit-content;">MELSEC</div> <table border="0" style="width: 100%;"> <tr><td>LD</td><td>M9036</td></tr> <tr><td>SET</td><td>M9053</td></tr> <tr><td>LD</td><td>TRUE</td></tr> <tr><td>DI</td><td></td></tr> </table>	LD	M9036	SET	M9053	LD	TRUE	DI			<table border="0" style="width: 100%;"> <tr><td>LD</td><td>M9036</td></tr> <tr><td>S</td><td>M9053</td></tr> <tr><td>LD</td><td>TRUE</td></tr> <tr><td>DI_M</td><td></td></tr> </table>	LD	M9036	S	M9053	LD	TRUE	DI_M	
LD	M9036																	
SET	M9053																	
LD	TRUE																	
DI																		
LD	M9036																	
S	M9053																	
LD	TRUE																	
DI_M																		
<div style="border: 1px solid black; padding: 5px; width: fit-content;">MELSEC</div> <table border="0" style="width: 100%;"> <tr><td>LD</td><td>M9036</td></tr> <tr><td>SET</td><td>M9053</td></tr> <tr><td>LD</td><td>TRUE</td></tr> <tr><td>EI</td><td></td></tr> </table>	LD	M9036	SET	M9053	LD	TRUE	EI			<table border="0" style="width: 100%;"> <tr><td>LD</td><td>M9036</td></tr> <tr><td>S</td><td>M9053</td></tr> <tr><td>LD</td><td>TRUE</td></tr> <tr><td>EI_M</td><td></td></tr> </table>	LD	M9036	S	M9053	LD	TRUE	EI_M	
LD	M9036																	
SET	M9053																	
LD	TRUE																	
EI																		
LD	M9036																	
S	M9053																	
LD	TRUE																	
EI_M																		

**Переменные**

Операнд	Значение	Тип данных
—	—	—

**Принцип действия****Условие выполнения для обновления канала связи****Общие сведения**

Обновление канала связи (см. "Команда COM") зависит от команды EI/DI. Функция команды EI/DI в процессоре AnN или A2C зависит от состояния специального маркера M9053. Лишь в том случае, если этот маркер установлен, эти команды служат в качестве условия обновления канала связи. Если маркер M9053 не установлен, эти команды служат в качестве условия для выполнения программы прерывания.

**DI Дезактивация обновления канала связи**

Команда DI предотвращает обновление канала связи до тех пор, пока в очереди команд программы не появится команда EI. После включения или сброса центрального процессора обновление разрешено.

Обновление канала возможно при каждой обработке команды END.

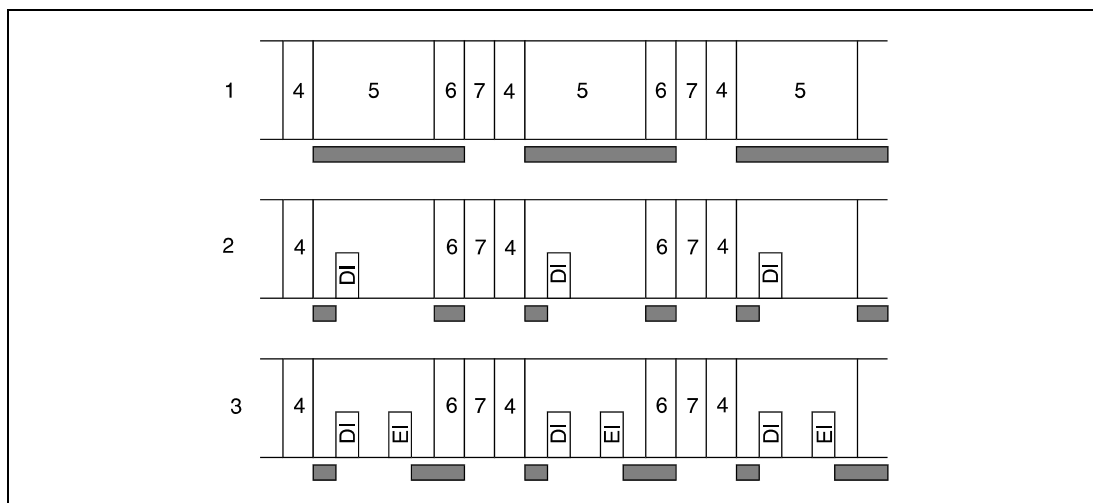
**EI Активация обновления канала связи**

Обновление канала становится возможным после установки команды EI.

**Условия выполнения**

На рисунке ниже показаны условия выполнения команды EI/DI.

Обработка сетевых данных возможна в заштрихованных областях. Если обработка происходит не с постоянной длительностью цикла, время ожидания постоянного цикла отпадает. При непосредственной обработке обновление входов и выходов не возможно.



1 выполнение программы без команды EI/DI

2 выполнение программы с командой DI

3 выполнение программы с командами EI/DI

4 обновление входов и выходов

5 основная программа

6 обработка команды END

7 время ожидания в случае программы с постоянной длительностью цикла

Обработка происходит при выполненном условии выполнения.

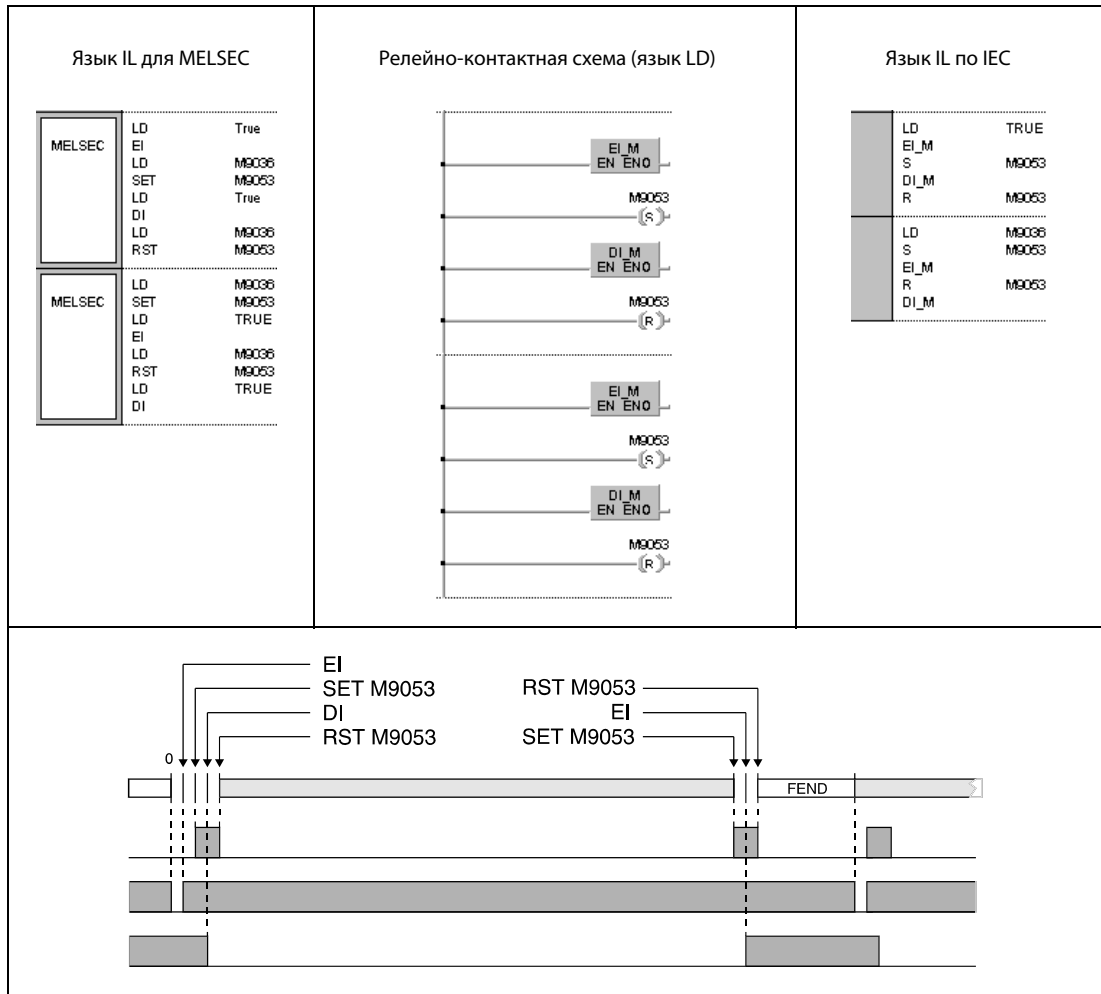
Функция команды EI/DI зависит от состояния специального маркера M9053. После выполнения команды EI/DI маркер M9053 можно включать или выключать.

Если команда EI или DI находится в пределах команды MC, то ее обработка происходит независимо от выполнения команды MC.

**Пример**

EI

Следующая программа предотвращает обновление канала до тех пор, пока не выполняется команда EI (незадолго до обработки команды FEND). Вызов программы прерывания возможен в любое время. На рисунке показана временная последовательность отработки программы.



## 6.8 Прочие команды

Перечисленные в таблице команды позволяют программировать специальные таймеры и счетчики, а также импульсные счетчики и генераторы импульсов. Кроме того, среди этих команд имеются команды для позиционирования поворотных столов и для образования матриц ввода.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC
Однофазный возрастающий/убывающий счетчик	UDCNT1	UDCNT1_M
Двухфазный возрастающий/убывающий счетчик	UDCNT2	UDCNT2_M
Программируемый таймер	TTMR	TTMR_M
Таймер для специальных функций	STMR	STMR_M
	STMRH	STMRH_M
Команда для позиционирования поворотных столов	ROTC	ROTC_M
Линейно изменяющийся сигнал	RAMP	RAMP_M
Счетчик импульсов	SPD	SPD_M
Импульсный выход с программируемой частотой и количеством импульсов	PLSY	PLSY_M
Широтно-импульсная модуляция	PWM	PWM_M
Образование матрицы ввода	MTR	MTR_M

### 6.8.1 UDCNT1

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

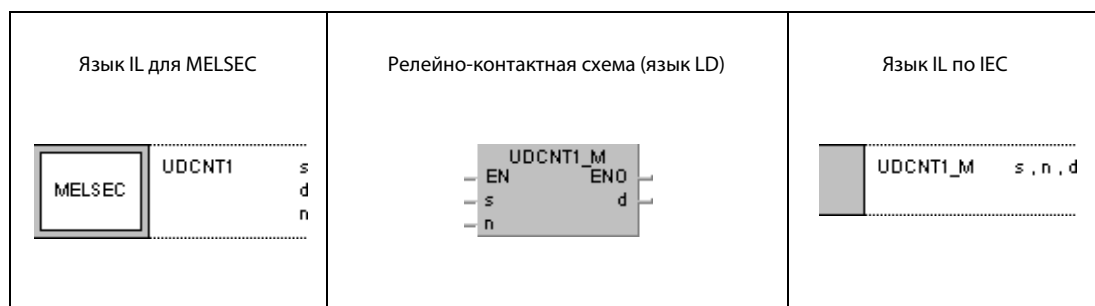
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s	● <sup>1</sup>	—	—	—	—	—	—	—	—	4	
d	—	● <sup>2</sup>	—	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

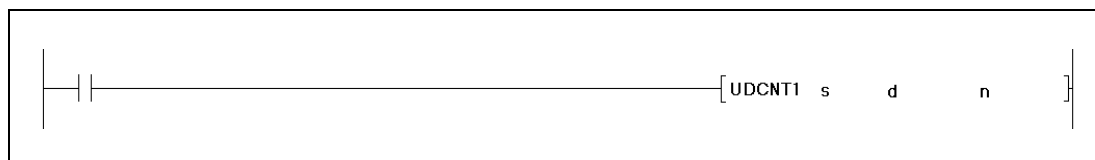
<sup>1</sup> только X

<sup>2</sup> только C

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных	
		MELSEC	IEC
s	s + 0: вход счета (импульсный сигнал, фаза)	бит	массив [1..2] переменных типа BOOL
	s + 1: установка направления счета 0 = возрастающий счет 1 = убывающий счет		
d	Счетчик, выполняющий команду UDCNT1.	BIN, 16 бит (только счетчик)	ANY16
n	заданное значение для контакта счетчика	BIN, 16 бит	ANY16

**Принцип действия****Однофазный возрастающий/убывающий счетчик****UDCNT1 Команда счетчика**

Если указанный в  $s + 0$  (`Array_s [0]`) вход изменяет свое состояние с 0 на 1, текущее состояние указанного в  $d$  счетчика обновляется. Это означает, что подсчитываются только нарастающие фронты импульсов.

Направление счета зависит от состояния входа, указанного в  $s + 1$  (`Array_s [1]`):

Если этот вход имеет состояние 0, то импульсы входа, указанного в  $s + 0$  (`Array_s [0]`), добавляются к текущему состоянию счетчика.

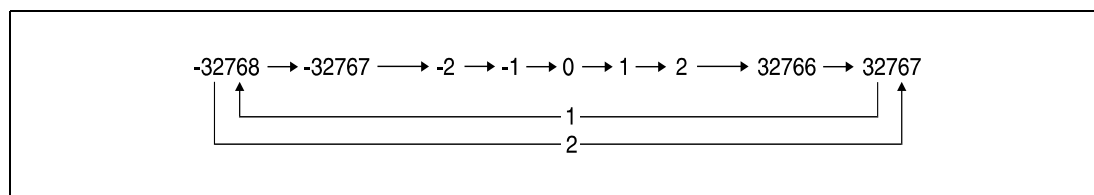
Если этот вход имеет состояние 1, то импульсы вычитаются из текущего состояния счетчика.

Процесс счета происходит следующим образом:

Контакт указанного в  $d$  счетчика включается (1), если в результате нарастания счета состояние счетчика совпало с состоянием, указанным в  $n$ . Процесс счета продолжается и при установленном контакте счетчика (см. пример).

При убывающем счете контакт счетчика снова сбрасывается при совпадении состояния счетчика со значением  $n - 1$  (см. пример).

Счетчик, указанный в  $d$ , представляет собой бесконечный счетчик. Если текущее состояние счетчика равно 32767 и повышается на 1, то счетчик перескакивает на -32768. При состоянии счетчика -32768 и дальнейшем уменьшении на 1 счетчик также перескакивает на 32767. Это наглядно изображено на следующей иллюстрации.



<sup>1</sup> при возрастающем счете

<sup>2</sup> при убывающем счете

Команда UDCNT1 запускается при включении условия выполнения и перестает действовать при выключении условия выполнения. При повторном включении счетчик возобновляет счет с того места, на котором он остановился.

Команда RST стирает содержимое указанного в  $d$  счетчика и выключает соответствующий контакт счетчика.

**ПРИМЕЧАНИЯ**

При использовании команды UDCNT1 процесс счета происходит во время прерывания центрального процессора (1 мс в случае многопроцессорного ЦПУ серии "Q" и 5 мс в случае процессоров QnA). Поэтому для правильного счета длительность импульса и время паузы между импульсами должны быть больше 1 мс или 5 мс (соответственно для этих процессоров).

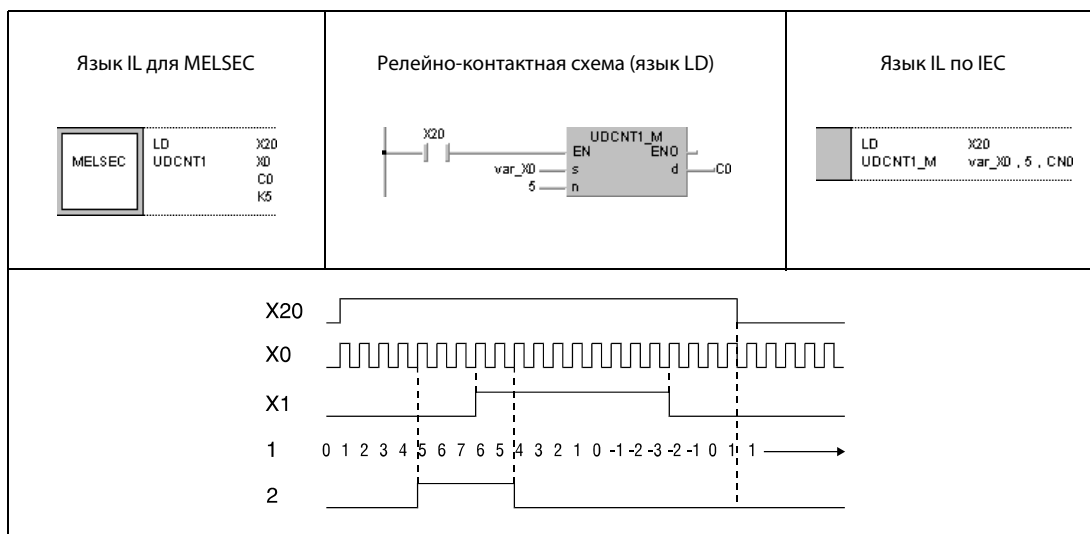
Настройки счетчиков не могут быть изменены во время процесса счета (вход, указанный в  $s + 0$  (`Array_s [0]`) имеет состояние 1). Чтобы сделать изменения, необходимо сбросить вход, указанный в  $s + 0$  (`Array_s [0]`), на 0.

Счетчик, используемый командой UDCNT1, невозможно одновременно использовать другими командами. При попытке такого использования правильный результат счета более не гарантируется.

**Пример**

UDCNT1

В следующей программе счетчик C0 (возрастающий/убывающий) применяется для подсчета положительных фронтов на входе X0 при включенном входе X20.



<sup>1</sup> состояние счетчика

<sup>2</sup> контакт счетчика C0

**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в раздел 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

## 6.8.2 UDCNT2

## Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

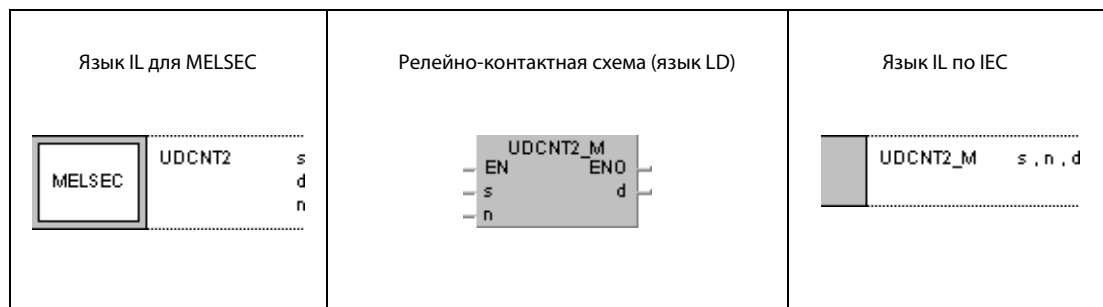
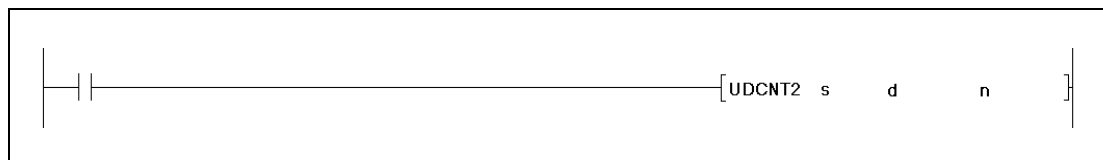
<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

Операнды  
MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s	● <sup>1</sup>	—	—	—	—	—	—	—	—	—	4
d	—	● <sup>2</sup>	—	—	—	—	—	—	—	—	
n	●	●	●	●	●	●	●	●	—	—	

<sup>1</sup> только X

<sup>2</sup> только C

GX IEC  
DeveloperGX  
Developer

## Переменные

Операнд	Значение	Тип данных	
		MELSEC	IEC
s	s + 0: вход счета (импульсный сигнал, фаза "А")	бит	массив [1..2] переменных типа BOOL
	s + 1: вход счета (импульсный сигнал, фаза "В")		
d	Счетчик, выполняющий команду UDCNT2.	BIN, 16 бит (только счетчик)	ANY16
n	Заданное значение (уставка) для контакта счетчика	BIN, 16 бит	ANY16



**Принцип действия****Двухфазный возрастающий/убывающий счетчик****UDCNT2      Команда счетчика**

Состояние указанного в *d* счетчика изменяется в зависимости от состояния двух входов, указанных в *s + 0* (*Array\_s [0]*) и *s + 1* (*Array\_s [1]*).

Направление счета определяется следующим образом:

Если вход, указанный в *s + 0* (*Array\_s [0]*), имеет состояние 1, то при изменении состояния входа, указанного в *s + 1* (*Array\_s [1]*), с 0 на 1 текущее состояние счетчика повышается на 1.

Если вход, указанный в *s + 0* (*Array\_s [0]*), имеет состояние 1, то при изменении состояния входа, указанного в *s + 1* (*Array\_s [1]*), с 1 на 0 текущее состояние счетчика уменьшается на 1.

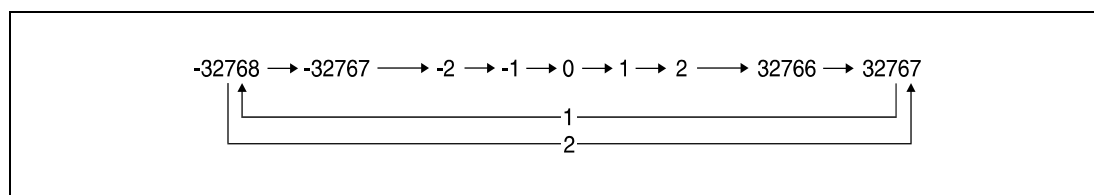
Если вход, указанный в *s + 0* (*Array\_s [0]*), имеет состояние 0, операция счета не выполняется.

Процесс счета происходит следующим образом:

Контакт указанного в *d* счетчика включается (1), если в результате нарастания счета состояние счетчика совпало с состоянием, указанным в *n*. Процесс счета продолжается и при установленном контакте счетчика (см. пример).

При убывающем счете контакт счетчика снова сбрасывается при совпадении состояния счетчика со значением *n - 1* (см. пример).

Счетчик, указанный в *d*, представляет собой бесконечный счетчик. Если текущее состояние счетчика равно 32767 и оно повышается на 1, счетчик перескакивает на -32768. При состоянии счетчика -32768 и дальнейшем уменьшении на 1 счетчик также перескакивает на 32767. Это наглядно изображено на следующей иллюстрации.



<sup>1</sup> при возрастающем счете

<sup>2</sup> при убывающем счете

Команда UDCNT2 запускается при включении условия выполнения и останавливается при отключении условия выполнения. При повторном включении счетчик возобновляет счет с того места, на котором он остановился.

Команда RST стирает содержимое указанного в *d* счетчика и выключает соответствующий контакт счетчика.

**ПРИМЕЧАНИЯ**

При использовании команды UDCNT2 процесс счета происходит во время прерывания центрального процессора (1 мс в случае многопроцессорного ЦПУ серии "Q" и 5 мс в случае процессоров QnA). Поэтому для правильного счета длительность импульса и время паузы между импульсами должны быть больше 1 мс или 5 мс (соответственно для названных процессоров).

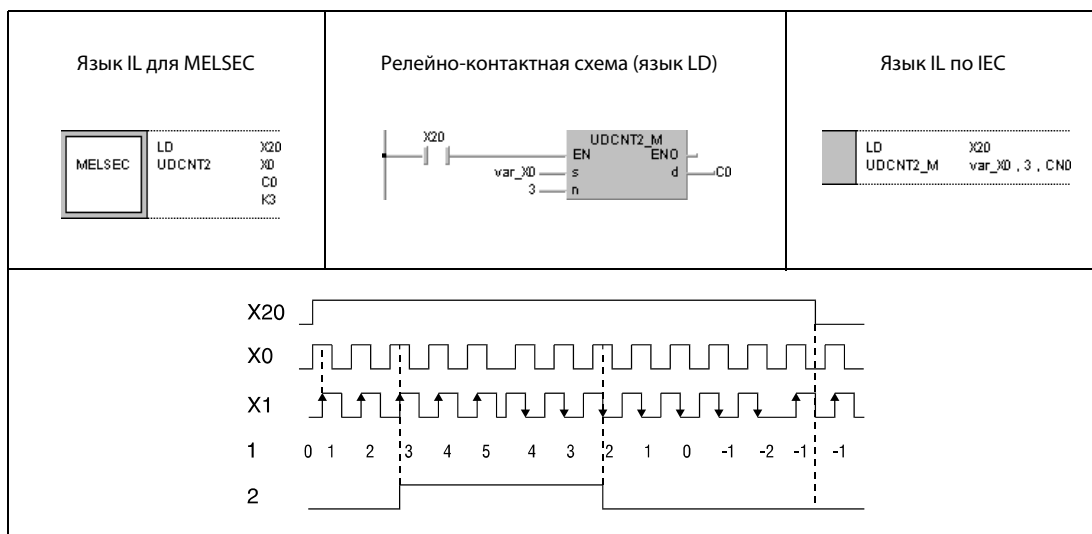
Во время счета настройки счетчика не могут быть изменены (вход, указанный в *s + 0* (*Array\_s [0]*), имеет состояние 1). Чтобы сделать изменения, необходимо сбросить вход, указанный в *s + 0* (*Array\_s [0]*), на 0.

Счетчик, используемый командой UDCNT2, невозможно одновременно использовать в других командах. При попытке такого использования правильный результат счета более не гарантируется.

**Пример**

## UDCNT2

Следующая программа ведет счет с помощью счетчика C0 после включения входа X20. Результат и направление счета (вверх/вниз) зависят от состояний X0 и X1.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 6.8.3 TTMR

**Процессор**

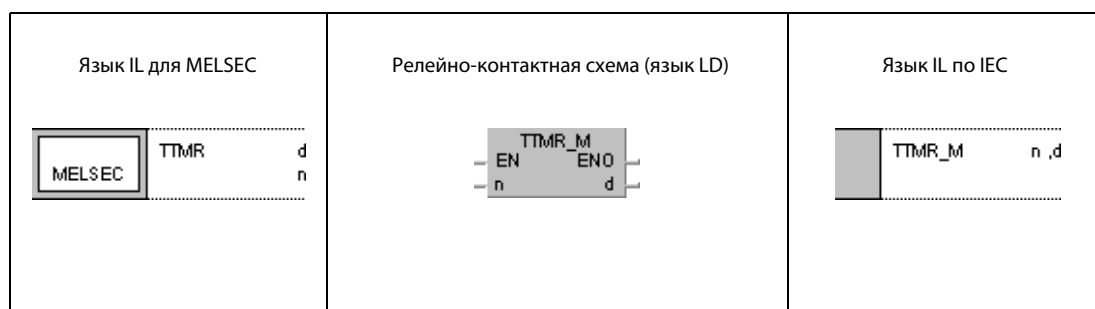
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

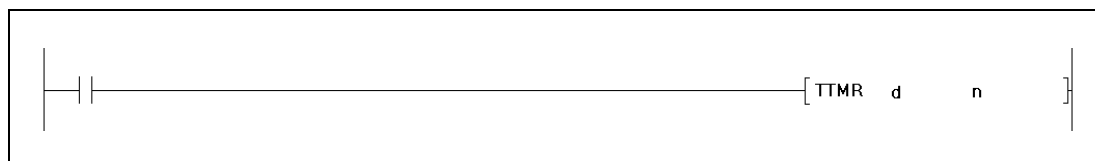
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непос. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
d	—	●	●	—	—	—	—	—	SM0	3	
n	—	●	●	●	●	●	●	—			

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных	
		MELSEC	IEC
d	d + 0: операнд, в котором сохраняется результат измерения	BIN, 16 бит	массив [1..2] переменных типа ANY16
	d + 1: для внутреннего использования в ЦПУ		
n	Множитель для результата измерения		ANY16

**Принцип действия** **Программируемый таймер**

**TTMR Команда таймера**

Таймер, программируемой с помощью команды TTMR, измеряет время в секундах на входе команды. Результат измерения умножается на указанный в n коэффициент и сохраняется в указанном в d операнде (Array\_d [0] + [1]).

При нарастающем фронте на входе команды содержимое d + 0 (Array\_d [0]) и d + 1 (Array\_d [1]) стирается.

Указание коэффициента в n:

n = 0, коэффициент 1

n = 1, коэффициент 10

n = 2, коэффициент 100

**ПРИМЕЧАНИЯ** *Время измеряется во время выполнения команды TTMR. Применение команды JMP или подобных команд в отношении команды TTMR приводит к ошибочному измерению времени.*

*Изменение указанного в n коэффициента во время выполнения команды TTMR искажает результат измерения.*

*Команду TTMR можно использовать и в программах с низкой скоростью обработки.*

*Указанное в d + 1 (Array\_d [1]) значение служит для внутрисистемной обработки и его не следует изменять. Изменение этого значения приводит к искажению результата, сохраненного в d + 0 (Array\_d [0]).*

**Источники ошибок** В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в n значение находится вне диапазона между 0 и 2 (код ошибки 4100).

**Пример** TTMR

Следующая программа измеряет длительность включенного состояния входа X0 в секундах (коэффициент 1) и сохраняет результат по адресу D0.



**ПРИМЕЧАНИЕ** *Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.*

### 6.8.4 STMR, STMRH

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

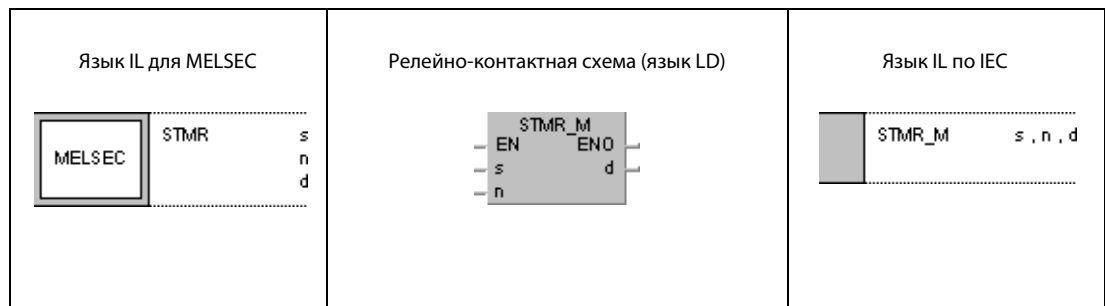
<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

**Операнды MELSEC Q**

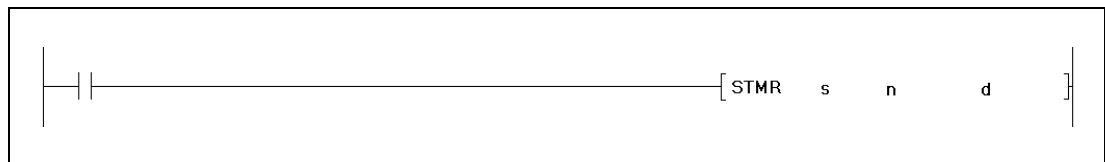
	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы			Иные
	битовые	словные		битовые	словные						
s	—	● <sup>1</sup>	—	—	—	—	—	—	—	3	
n	●	—	—	—	—	—	—	—	—		
d	—	●	●	●	●	●	●	●	—		

<sup>1</sup> Можно использовать только для выбора таймера (T).

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных	
		MELSEC	IEC
s	Номер таймера	BIN, 16 бит (только таймер)	ANY16
n	Уставка времени	BIN, 16 бит	ANY16
d	d + 0: выход таймера с задержкой выключения d + 1: выход таймера с устанавливающим импульсом (установка по отрицательному фронту команды) d + 2: выход таймера с устанавливающим импульсом (установка по положительному фронту команды) d + 3: выход таймера с задержкой включения	бит	массив [1..4] переменных типа BOOL

<b>Принцип действия</b>	<b>Таймер для специальных функций</b>	
	<b>STMR</b>	<b>Команда медленного (низкоскоростного) таймера</b>
	<b>STMRH</b>	<b>Команда быстрого (высокоскоростного) таймера</b>

Команда STMR использует выходы, указанные с  $d + 0$  (Array\_d [0]) по  $d + 3$  (Array\_d [3]), для реализации четырех различных функций таймера:

Выход таймера с задержкой выключения ( $d + 0$  (Array\_d [0]))

Выход, указанный в  $d + 0$  (Array\_d [0]), устанавливается при нарастающем фронте условия выполнения и сбрасывается с задержкой после отрицательного фронта условия выполнения. Время задержки указано с помощью  $n$ .

Выход таймера с устанавливающим импульсом (установка при отрицательном фронте команды,  $d + 1$  (Array\_d [1]))

Выход, указанный в  $d + 1$  (Array\_d [1]), устанавливается при отрицательном фронте условия выполнения. Выход сбрасывается по истечении времени, указанного с помощью  $n$ , или при положительном фронте условия выполнения.

Выход таймера с устанавливающим импульсом

(установка при положительном фронте условия выполнения,  $d + 2$  (Array\_d [2]))

Выход, указанный в  $d + 2$  (Array\_d [2]), устанавливается при нарастающем фронте условия выполнения. Сброс происходит по истечении времени, указанного с помощью  $n$ , или при отрицательном фронте условия выполнения.

Выход таймера с задержкой включения ( $d + 3$  (Array [3]))

Выход, указанный в  $d + 3$  (Array\_d [3]), устанавливается при отрицательном фронте катушки таймера. Это соответствует задержке включения на время, указанное с помощью  $n$ . Выход устанавливается также при отрицательном фронте условия выполнения и снова сбрасывается лишь по истечении времени, установленного с помощью  $n$ .

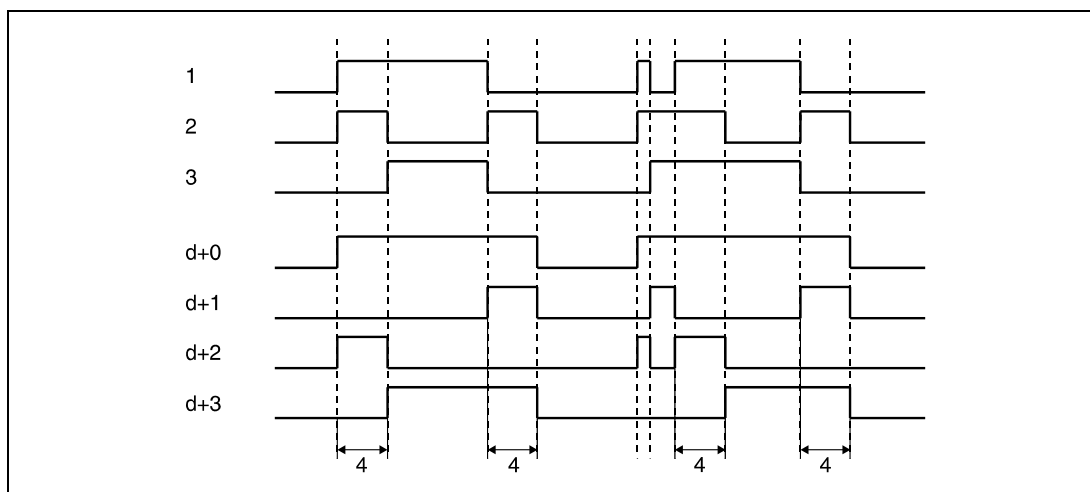
Катушка указанного в  $s$  таймера устанавливается при нарастающем фронте условия выполнения и начинает отсчет времени, указанного с помощью  $n$ .

Катушка таймера измеряет время до того момента, в который измеренное время совпадает с временем, указанным с помощью  $n$ , после чего она отпускает контакт.

Если до истечения времени, указанного с помощью  $n$ , условие выполнения перестало выполняться, катушка таймера остается установленной, но измерение времени приостанавливается.

При повторном включении условия выполнения результат измерения стирается и измерение времени начинается заново.

Контакт указанного в  $s$  таймера устанавливается либо при установленной катушке таймера и отрицательном фронте условия выполнения, либо при установленном условии выполнения и отрицательном фронте катушки. Контакт таймера сбрасывается при неустановленной катушке и отрицательном фронте условия выполнения. Контакт таймера служит только для внутри системной обработки в центральном процессоре.



- <sup>1</sup> условие выполнения
- <sup>2</sup> катушка указанного в s таймера
- <sup>3</sup> контакт указанного в s таймера
- <sup>4</sup> Время, указанное в n.

Измерение истекшего времени происходит во время выполнения команды SMTR. Если в отношении команды SMTR применена команда JMP или подобная ей команда, правильное измерение времени более не возможно.

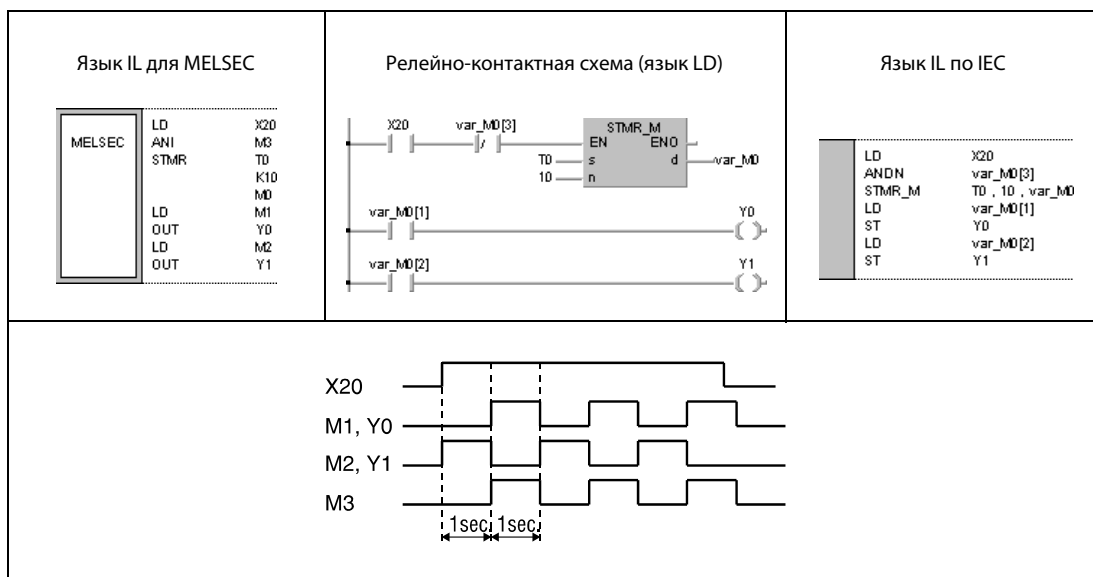
Фактические значения времени для указанных в d выходов таймера вычисляются путем умножения настройки n на единицу измерения для медленных таймеров (стандартная настройка: 100 мс).

Указываемая в n константа может находиться в диапазоне между 1 и 32767.

Указанный в s таймер нельзя использовать в команде OUT. Если команда OUT и команда SMTR используют один и тот же таймер, команда SMTR не может выполняться безошибочно.

**Пример** SMTR

Следующая программа при включении входа X20 попеременно включает выходы Y0 и Y1 на 1 секунду. Здесь используется 100-миллисекундный таймер. Длительность включения (одна секунда) рассчитывается как константа K10, умноженная на 100 мс.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.



### 6.8.5 ROTC

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

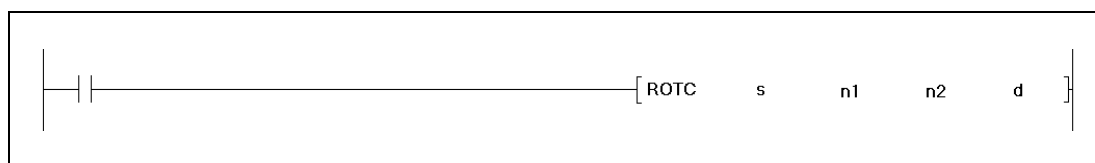
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□NG□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	SM0	5	
n1	●	●	●	●	●	●	●	—			
n2	●	●	●	●	●	●	●	—			
d	●	—	—	—	—	—	—	—			

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных	
		MELSEC	IEC
s	s + 0: измерение частоты вращения (только для внутрисистемного использования)	BIN 16 битов	массив [1..3] данных типа ANY16
	s + 1: номер позиции		
	s + 2: номер сектора		
n1	количество делений (от 2 до 32767)		ANY16
n2	количество делений для низкой скорости вращения (ползучей скорости) (от 0 до n1).		ANY16

Переменные

Операнд	Значение	Тип данных	
		MELSEC	IES
d	d + 0: входной сигнал (фаза "A")	бит	массив [1..8] переменных типа Bool
	d + 1: входной сигнал (фаза "B")		
	d + 2: входной сигнал определения нулевой точки		
	d + 3: выходной сигнал "вперед, высокая частота вращения" (внутрисистемный)		
	d + 4: выходной сигнал "вперед, низкая частота вращения" (внутрисистемный)		
	d + 5: выходной сигнал "стоп" (внутрисистемный)		
	d + 6: выходной сигнал "назад, высокая частота вращения" (внутрисистемный)		
	d + 7: выходной сигнал "назад, низкая частота вращения" (внутрисистемный)		

**Принцип действия****Команда позиционирования поворотных столов****ROTC Команда позиционирования**

Команда ROTC позволяет позиционировать поворотный стол с указанным в  $n1$  количеством секторов (делений) таким образом, чтобы указанный в  $s + 2$  ( $Array\_s [2]$ ) сектор переместился в указанную в  $s + 1$  ( $Array\_s [1]$ ) позицию.

Позиции и сектора на поворотном столе нумеруются против часовой стрелки.

Значение, указанное в  $s + 0$  ( $Array\_s [0]$ ), используется внутри системы для определения позиции секторов относительно позиции 0. Значение, указанное в  $s + 0$  ( $Array\_s [0]$ ), изменять нельзя, так как иначе позиционирование станет невозможным.

Значение, указанное в  $n2$ , означает количество секторов, на которое поворотный стол перемещается с низкой (ползучей) скоростью. Это число должно быть меньше или равно числу  $n1$  (количество делений).

Входы, указанные в  $d + 0$  ( $Array\_d [0]$ ) и  $d + 1$  ( $Array\_d [1]$ ) (фаза "A"/"B"), служат для определения направления вращения. Оба входа управляются импульсами. Если вход, указанный в  $d + 0$  ( $Array\_d [0]$ ) (фаза "A"), имеет состояние 1, то направление вращения определяется фронтом импульса на входе, указанном в  $d + 1$  ( $Array\_d [1]$ ) (фаза B), следующим образом:

Если в этот момент фаза "B" имеет положительный фронт, вращение происходит по часовой стрелке (прямое вращение).

Если в этот момент фаза "B" имеет отрицательный фронт, вращение происходит против часовой стрелки (обратное вращение).

Вход, указанный в  $d + 2$  ( $Array\_d [2]$ ), служит для определения нулевой точки. Этот вход устанавливается, если сектор 0 достиг позиции 0. Если этот вход установлен во время выполнения команды ROTC, значение в  $s + 0$  ( $Array\_s [0]$ ) сбрасывается. Для безупречного функционирования рекомендуется выполнять эту операцию сброса значения, указанного в  $s + 0$  ( $Array\_s [0]$ ), перед самим позиционированием с помощью команды ROTC.

Данные с  $d + 3$  ( $Array\_d [3]$ ) по  $d + 7$  ( $Array\_d [7]$ ) содержат выходные сигналы для управления поворотным столом. Устанавливаемый выходной сигнал зависит от текущего результата выполнения команды ROTC.

Если непосредственно перед выполнением команды ROTC все результаты операций были равны 0, то выходы, указанные с  $d + 3$  ( $Array\_d [3]$ ) по  $d + 7$  ( $Array\_d [7]$ ), сбрасываются без выполнения позиционирования. Эти выходы сбрасываются также при отключении условия выполнения.

Команду ROTC можно использовать в программе только один раз. Многократное использование этой команды приводит к ее ошибочному выполнению.

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Значение, указанное с  $s + 0$  ( $Array\_s [0]$ ) по  $s + 2$  ( $Array\_s [2]$ ) или в  $n2$ , превышает значение, указанное в  $n1$  (код ошибки 4100).

**Пример**

ROTC

Следующая программа через контакты X0, X1 (инкрементальный энкодер) и X2 обращается к маркерам для определения направления вращения и нулевой точки с M0 (var\_M0 Array [0]) по M2 (var\_M0 Array [2]). Контакт X2 активируется, если сектор 0 находится в позиции 0 (определение нулевой точки).

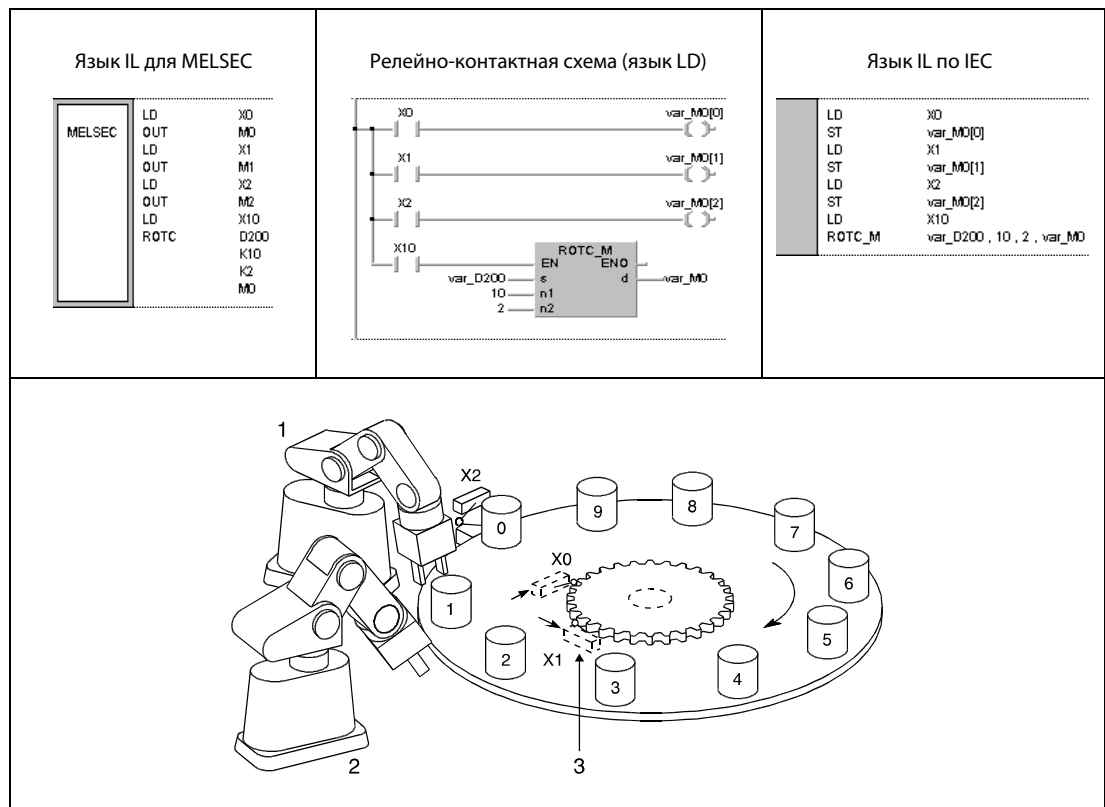
Изображенный здесь поворотный стол подразделен на 10 секторов.

Перед выполнением команды ROTC в регистрах D201 (var\_D200 Array [1]) и D202 (var\_D200 Array [2]) следует указать, какая деталь (сектор) движется к какой станции (позиции).

Если указано n1 = 10, то при одном обороте поворотного стола контакт регистра счета выдает 10 импульсов (делений). В n2 = 2 указано количество делений, которое стол проходит на ползучей скорости.

Например, если в регистре D201 (var\_D200 Array [1]) указано значение 0, а в регистре D202 (var\_D200 [2]) – значение 3, то поворотный стол кратчайшим путем (вращаясь вправо) доставляет деталь 3 (сектор 3) к станции 0 (позиции 0), при этом путь от сектора 1 до сектора 3 проходит на ползучей скорости.

Взаимосвязь между отдельными регистрами, маркерами или элементами массива с соответствующими функциями пояснена в таблице под примером программы.



- 1 позиция 0
- 2 позиция 1
- 3 инкрементальный энкодер

Регистр данных	Описание	Примечание
D200 (var_D200 Array [0])	регистр счета	
D201 (var_D200 Array [1])	позиция захвата (станции)	Перед этим с помощью команды MOV значения записываются в регистры данных с D200 (var_D200 Array [0]) по D202 (var_D200 Array [2]).
D202 (var_D200 Array [2])	позиция требуемой детали	
M0 (var_M0 Array [0])	сигнал фазы "А"	Маркеры с M0 (var_M0 Array [0]) по M2 (var_M0 Array [2]) устанавливаются через входы с X0 по X2 (см. пример программы).
M1 (var_M0 Array [1])	сигнал фазы "В"	
M2 (var_M0 Array [2])	сигнал нулевой точки	
M3 (var_M0 Array [3])	вращение вперед (высокая скорость)	
M4 (var_M0 Array [4])	вращение вперед (ползучая скорость)	После включения X10 активируется команда ROTC, и маркерам с M3 (var_M0 Array [3]) по M7 (var_M0 Array [7]) автоматически присваиваются неизменные функции. После отключения X10 эти маркеры сбрасываются.
M5 (var_M0 Array [5])	стоп	
M6 (var_M0 Array [6])	вращение назад (высокая скорость)	
M7 (var_M0 Array [7])	вращение назад (ползучая скорость)	

**ПРИМЕЧАНИЕ** *Без объявления переменных в заголовке РОУ этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.*

### 6.8.6 RAMP

Процессор

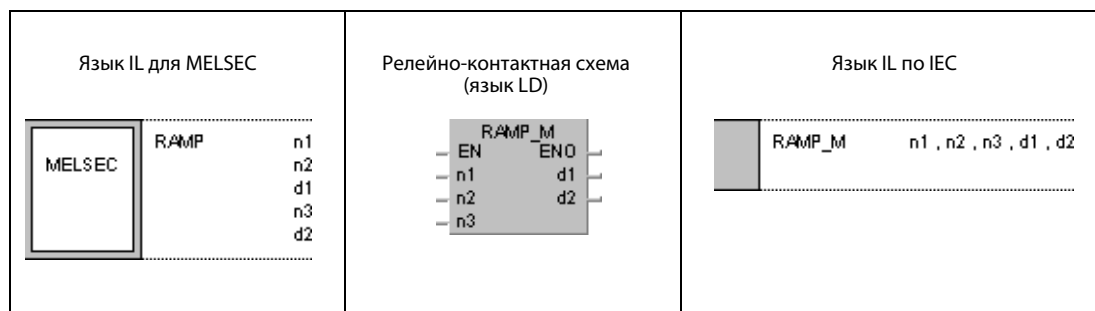
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

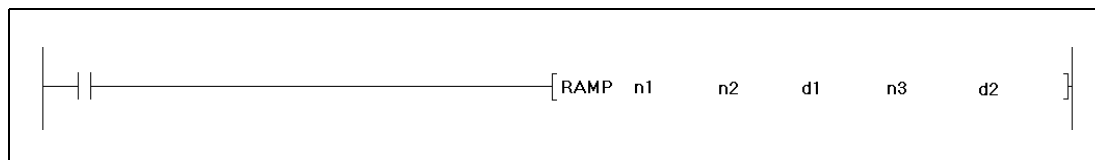
Операнды  
MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
n1	●	●	●	●	●	●	●	●	—	—	6
n2	●	●	●	●	●	●	●	●	—	—	
d1	●	●	●	●	●	●	—	—	—	—	
n3	●	●	●	●	●	●	●	●	—	—	
d2	●	—	—	—	—	—	—	—	—	—	

GX IEC  
Developer



GX  
Developer



Переменные

Операнд	Значение	Тип данных	
		MELSEC	IEC
n1	Начальное значение операции	BIN, 16 бит	ANY16
n2	Конечное значение операции		ANY16
d1	(d1) + 0: операнд, в котором сохранено текущее значение		массив [1..2] данных типа ANY16
	(d1) + 1: операнд, в котором сохранено количество выполненных шагов (только для внутрисистемного использования)		
n3	Количество шагов, которое требуется выполнить	ANY16	
d2	(d2) + 0: бит, устанавливаемый после полного выполнения команды.	бит	массив [1..2] переменных типа Bool
	(d2) + 1: бит, от состояния которого зависит сохранение результата операции		

**Принцип действия**

**Линейно изменяющийся сигнал**

**RAMP Команда для ступенчатого повышения содержимого памяти**

Команда RAMP ступенчато повышает содержимое  $(d1) + 0$  (Array\_d1 [0]) от начального значения, указанного в n1, до конечного значения, указанного в n2.

В n3 указано количество шагов, за которое должно быть выполнено вышеуказанное повышение.

В  $(d1) + 1$  (Array\_d1 [1]) сохраняется количество уже выполненных шагов для внутрисистемных нужд.

По окончании этой операции устанавливается операнд, указанный в  $(d2) + 0$  (Array\_d2 [0]).

Состояние операнда, указанного в  $(d2) + 0$  (Array\_d2 [0]), и содержимое  $(d1) + 0$  (Array\_d1 [0]) зависят от состояния операнда, указанного в  $(d2) + 1$  (Array\_d2 [1]), следующим образом:

Если операнд, указанный в  $(d2) + 1$  (Array\_d2 [1]), не установлен, то при следующем шаге операнд, указанный в  $(d2) + 0$  (Array\_d2 [0]), сбрасывается, и команда RAMP выполняет повторно повышение последнее значение, сохраненное в  $(d1) + 0$  (Array\_d1 [0]).

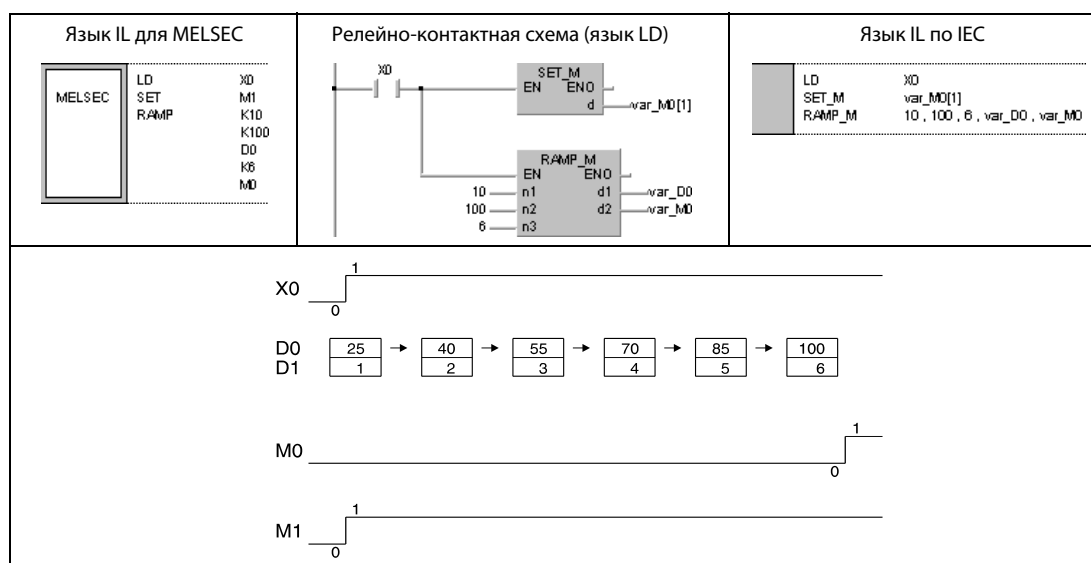
Если операнд, указанный в  $(d2) + 1$  (Array\_d2 [1]), установлен, то остается установленным и операнд, указанный в  $(d2) + 0$  (Array\_d2 [0]), и указанное в  $(d1) + 0$  (Array\_d1 [0]) значение не изменяется (сохранение). Если во время операции условие выполнения изменилось на 0, то содержимое  $(d1) + 0$  (Array\_d1 [0]) не изменяется. Если условие выполнения снова выполняется, то команда RAMP повышает содержимое  $(d1) + 0$  (Array\_d1 [0]), сохраненное перед сбросом условия выполнения.

Во время выполнения команды нельзя изменять значения в n1 и n2.

**Пример**

**RAMP**

Следующая программа повышает содержимое D0 за 6 шагов от 10 до 100 и по окончании операции сохраняет содержимое D0.



**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию можно найти в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

### 6.8.7 SPD

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

**Операнды MELSEC Q**

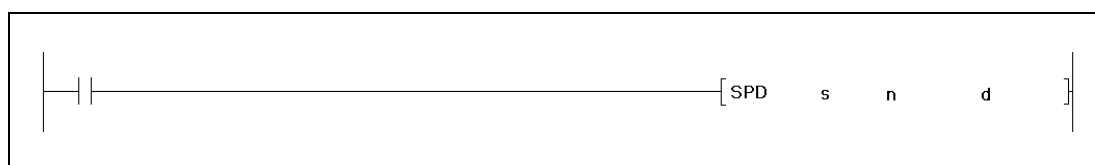
	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы			Иные
	битовые	словные		битовые	словные						
s	● <sup>1</sup>	—	—	●	●	●	●	●	—	—	4
n	●	●	●	—	—	—	—	—	—	—	
d	—	●	●	●	●	●	●	●	—	—	

<sup>1</sup> только X

**GX IEC Developer**



**GX Developer**



**Переменные**

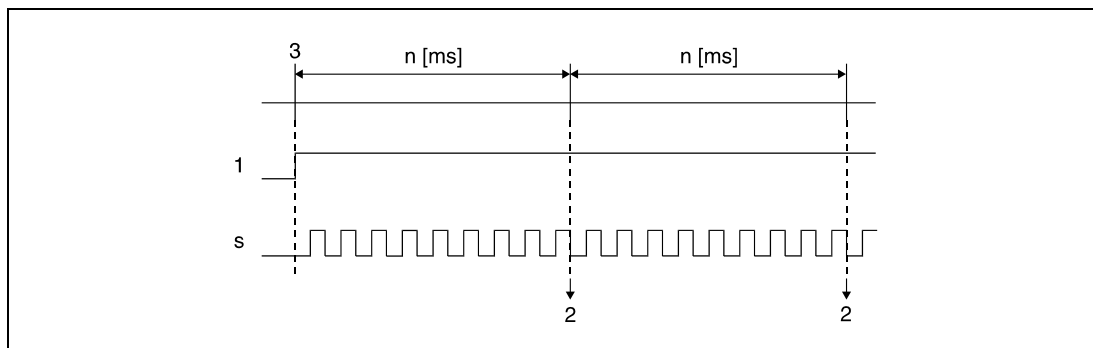
Операнд	Значение	Тип данных
s	Импульсный входной сигнал	бит
n	Длительность измерения в мс	BIN, 16 бит
d	Начальный адрес, начиная с которого сохраняется результат измерения.	



**Принцип действия**

**Счетчик импульсов**  
**SPD      Команда счета импульсов**

Команда SPD считает импульсы указанного в s входа на протяжении времени, указанного в n, и сохраняет результат измерения в операнде, указанном в d.



- 1 условие выполнения
- 2 результат сохраняется в d
- 3 начало измерения

До тех пор, пока условие выполнения установлено, по истечении времени измерения измерение снова начинается с 0. Чтобы остановить измерение, выполняемое командой SPD, необходимо сбросить условие выполнения.

**ПРИМЕЧАНИЯ**

Команда SPD сохраняет данные операндов в рабочей области центрального процессора и выполняет измерение во время системных прерываний. Чтобы распознавались все импульсы, их длительность должна быть равной или превышать время между системными прерываниями. В случае многопроцессорных ЦПУ серии "Q" это время равно 1 мс, а в случае процессоров QnA прерывание активируется каждые 5 мс.

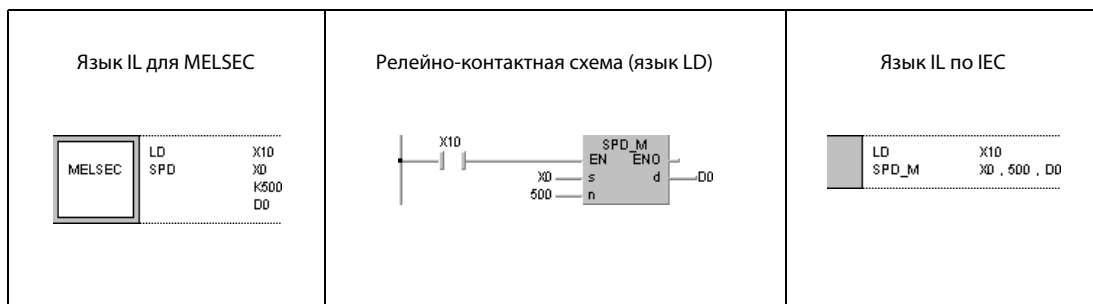
Команда не выполняется, если в случае применения процессора "Q" значение n = 0 или в случае применения процессоров QnA значение n = 0 или не кратно 5.

Команду SPD можно использовать во всей программе максимум шесть раз. Седьмая и все последующие команды SPD не выполняются.

**Пример**

**SPD**

Следующая программа после включения входа X10 измеряет импульсы входа X0 на протяжении 500 мс и сохраняет результат по адресу D0.



### 6.8.8 PLSY

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

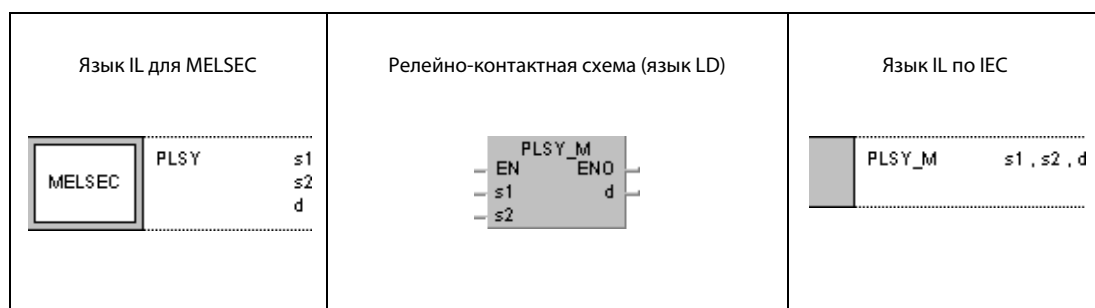
<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

**Операнды MELSEC Q**

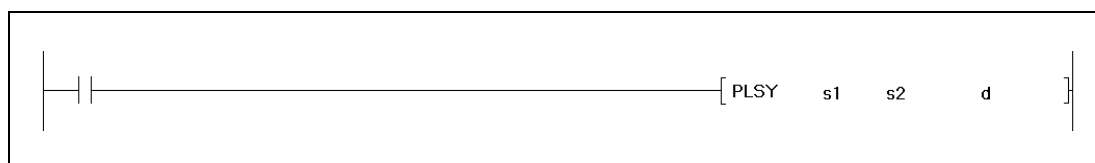
	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы	Иные		
	битовые	словные		битовые	словные						
s1	●	●	●	●	●	●	●	●	—	—	4
s2	●	●	●	●	●	●	●	●	—	—	
d	● <sup>1</sup>	—	—	—	—	—	—	—	—	—	

<sup>1</sup> только Y

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s1	Операнд, в котором сохранена частота.	BIN, 16 бит
s2	Операнд, в котором сохранено количество выходных импульсов.	
d	Операнд, в котором сохранен выход.	бит

**Принцип действия**      **Вывод программируемого количества импульсов**  
**PLSY      Команда вывода импульсов**

Команда PLSY выводит указанное в s2 количество импульсов с указанной в s1 частотой на указанный в d выход.

Указанная в s1 частота может быть в диапазоне между 1 и 100 Гц. Если в s1 введено значение 0, команда PLSY выдает непрерывный сигнал.

Указанное в s2 количество импульсов может находиться в диапазоне между 1 и 32767.

В d можно указывать только выходы, сопоставленные модулям вывода.

Вывод импульсов начинается при нарастающем фронте условия выполнения команды PLSY. Во время вывода импульсов условие выполнения нельзя сбрасывать. При сбросе условия выполнения вывод импульсов прекращается.

**ПРИМЕЧАНИЕ**      *Команда PLSY сохраняет данные операндов в рабочей области центрального процессора. Обновление выданных командой PLSY импульсов происходит во время системных прерываний. Чтобы распознавались все импульсы, их длительность должна быть равной или превышать время между системными прерываниями. В случае многопроцессорных ЦПУ серии "Q" это время равно 1 мс, а в случае процессоров QnA прерывание активируется каждые 5 мс.*

*Если условие выполнения выполнено, команду PLSY нельзя изменять. В противном случае вывод импульсов не удастся прекратить сбросом условия выполнения.*

*Команду PLSY можно использовать только один раз во всей программе.*

**Пример**      PLSY

Следующая программа после включения X0 выводит пять импульсов с частотой 10 Гц на выход Y20.



### 6.8.9 PWM

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

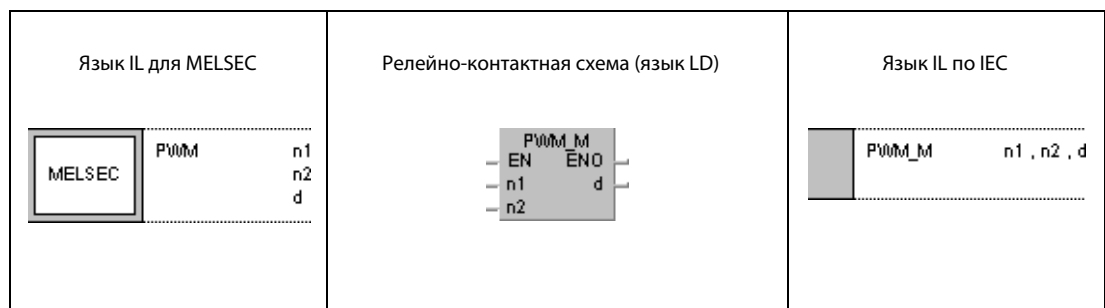
<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

Операнды  
MELSEC Q

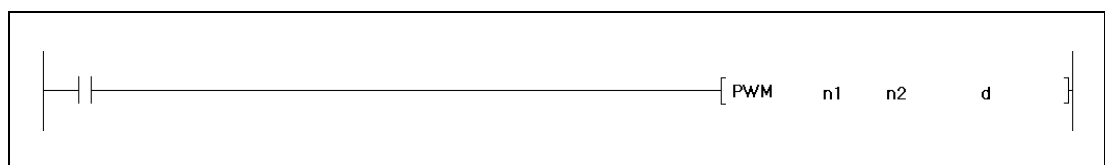
	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□NG□	Индексные регистры Zn	Константы	Иные		
	битовые	словные		битовые	словные						
n1	●	●	●	●	●	●	●	●	—	—	4
n2	●	●	●	●	●	●	●	●	—	—	
d	● <sup>1</sup>	—	—	—	—	—	—	—	—	—	

<sup>1</sup> только Y

GX IEC  
Developer



GX  
Developer

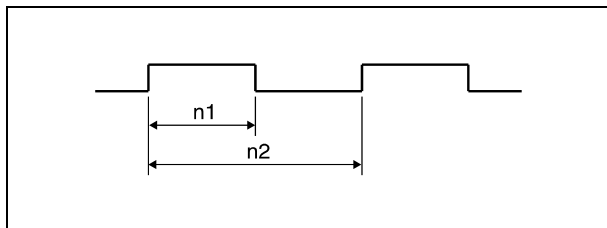


Переменные

Операнд	Значение	Тип данных
n1	Операнд, в котором сохранена длительность импульса.	BIN, 16 бит
n2	Операнд, в котором сохранена длительность периода.	
d	Операнд, на который выводится импульс.	бит

**Принцип действия**      **Широтно-импульсная модуляция**  
**PWM**      **Модулирующая команда**

Команда PWM выводит импульсы с длительностью периода, указанной в n2, и длительностью импульса, указанной в n1, на выход, указанный в d.



В случае многопроцессорных ЦПУ серии "Q" указанные в n1 и n2 значения могут находиться в диапазоне между 1 и 65535 мс, а в случае процессоров QnA – в диапазоне между 5 и 65535 мс. Значение n1 должно быть меньше значения n2.

**ПРИМЕЧАНИЕ**      Команда PWM сохраняет данные операндов в рабочей области центрального процессора и выполняет операции вывода во время системных прерываний (1 мс в случае многопроцессорных ЦПУ серии "Q" и 5 мс в случае процессоров QnA). Команду PWM можно использовать только один раз во всей программе.

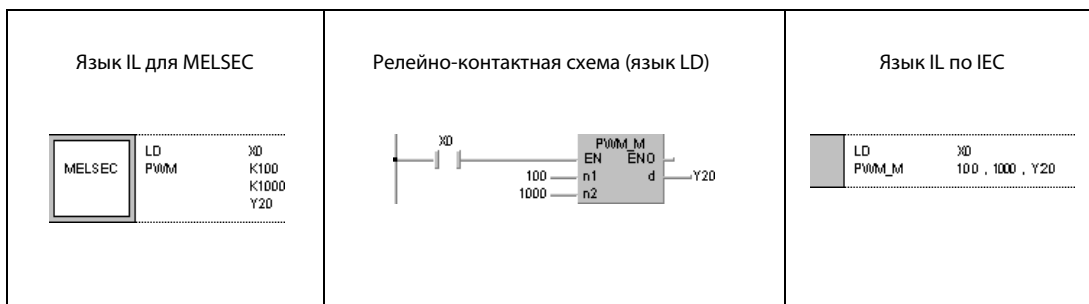
В следующих случаях команда PWM не выполняется:

- если n1 и n2 равны нулю.
- если n2 равен или меньше n1.
- если в случае применения процессора QnA значения n1 и n2 не кратны 5.

Во время выполнения команды PWM нельзя изменять значения n1, n2 и d.

**Пример**      PWM

Следующая программа после включения X0 выдает импульсы с длительностью периода 1 секунда и длительностью импульса 100 мс на выход Y20.



## 6.8.10 MTR

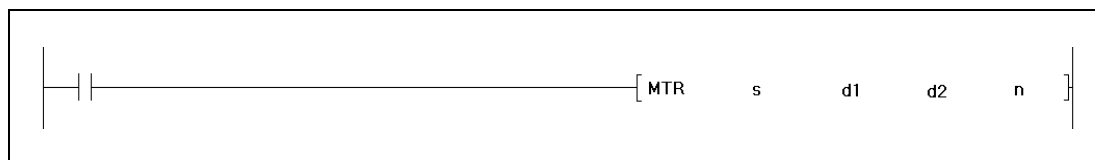
## Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы			Иные
	битовые	словные		битовые	словные						
s	●	—	—	—	—	—	—	—	SM0	5	
d1	●	—	—	—	—	—	—				
d2	●	—	—	—	—	—	—				
n	●	●	●	●	●	●	●	—			

GX IEC  
DeveloperGX  
Developer

## Переменные

Операнд	Значение	Тип данных
s	Адрес первого входа	бит
d1	Адрес первого выхода	
d2	Начальный адрес, начиная с которого в виде матрицы сохраняются входные данные.	
n	Количество опросов входов	BIN, 16 бит

**Принцип действия****Образование матрицы ввода****MTR Команда для n-кратного считывания информации с образованием матрицы ввода**

Команда MTR считывает информацию 16 битов (0/1), начиная с операнда, указанного в s. Количество повторений этого процесса (число рядов) указано в n. Состояния считанных битов сохраняются, начиная с операнда, указанного d2. В результате этого образуется матрица "16 битов x n рядов".

За каждый шаг могут быть считаны 16 битов.

Считывание непрерывно повторяется от первого до n-ного ряда.

В связи с тем, что команда MTR образует матрицу размером 16 битов на n рядов, указанный в d2 операнд должен быть способен вместить 16 битов x n рядов.

Отдельные ряды выбираются начиная с выхода, указанного в d1. Выход, соответствующий считываемому 16-битовому ряду, система устанавливает и сбрасывает автоматически. Количество выходов равно количеству рядов. Благодаря этому система может точно адресовать каждый отдельный ряд.

Операнды, указанные в s, d1 и d2, должны иметь только адреса, кратные 16.

Количество рядов может быть в диапазоне между 2 и 8.

Следует иметь в виду, что команда MTR работает непосредственно с текущими данными входов и выходов.

**Источники ошибок**

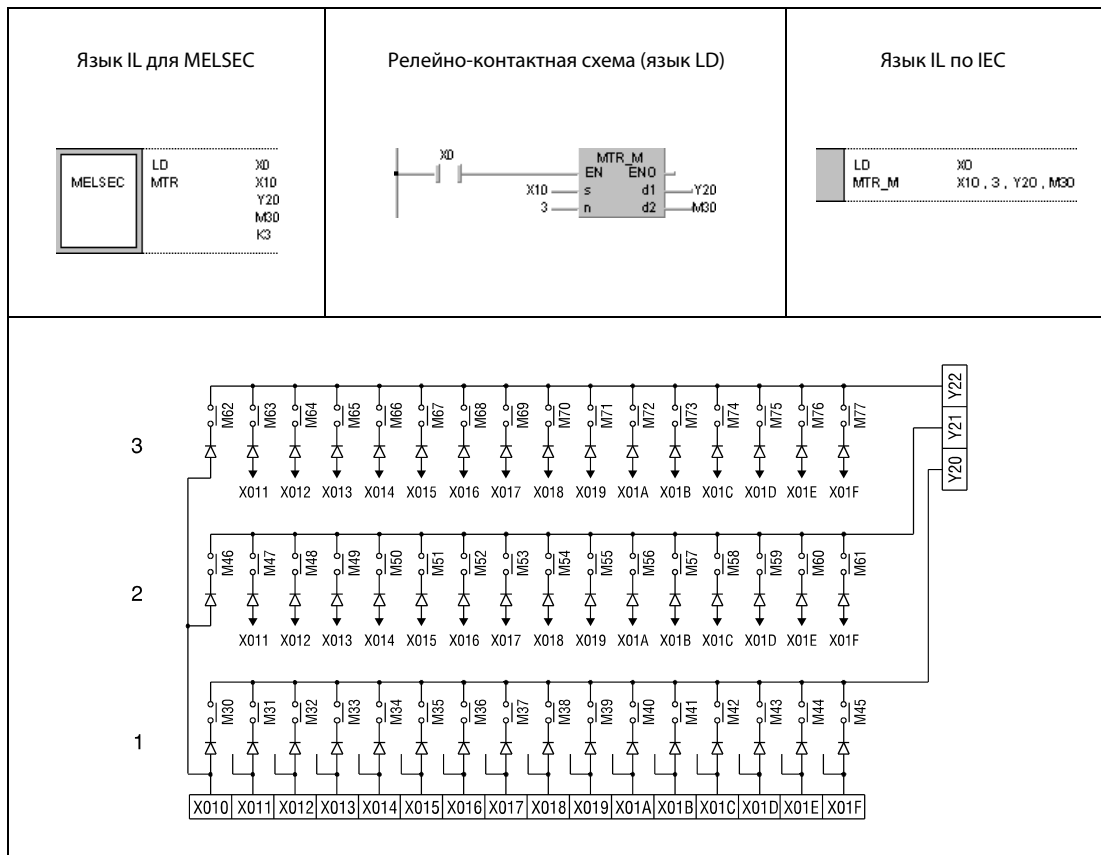
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Адреса операндов, указанных в s, d1 и d2, не кратны 16 (код ошибки 4101).
- Указанный в s операнд находится вне текущей области ввода (код ошибки 4101).
- Указанный в d1 операнд находится вне текущей области вывода (код ошибки 4101).
- Матрица размером "16 битов на n рядов" выходит за пределы области d2, предусмотренной для сохранения (код ошибки 4101).
- Указанное значение n находится вне диапазона 2...8 (код ошибки 4100).

Пример

MTR

Следующая программа после включения входа X0 считывает входы с X10 по X1F три раза и сохраняет результаты в маркерах с M30 по M77. В результате этого образуется матрица размером 16 битов x 3 ряда. Ряды адресуются через выходы Y20...Y22.



- <sup>1</sup> 1-й ряд
- <sup>2</sup> 2-й ряд
- <sup>3</sup> 3-й ряд



## 7

## Прикладные команды, часть II

Во втором разделе, посвященном прикладным командам, рассматриваются специфические прикладные команды для ряда особых функций. Подразделение этих команд на отдельные функции пояснено в следующей обзорной таблице.

Команды	Значение
Команды логических функций	логика И/ИЛИ, логика "исключающее ИЛИ/НЕ ИЛИ"
Команды вращения	вращение 16-битных и 32-битных данных вправо/влево
Команды сдвига	побитовый или поблочный сдвиг в пределах слова данных
Команды обработки битов	установка и сброс битов, опрос битов
Команды обработки данных	поиск данных в определенных областях, кодировка и декодировка данных, разъединение и объединение значений данных
Команды структурирования программы	команда повторения, вызов подпрограммы, вызов подпрограммы в другом программном файле, переключение между областями Main и Sub программы, вызов программы микрокомпьютера, индексная адресация части программы, сохранение индексированных адресов операндов в индексном перечне
Команды для работы с перечнями данных	запись данных в перечень для их дальнейшей обработки и считывание данных из перечня, стирание или вставление определенных блоков данных в перечень данных
Команды обращения к буферной памяти	обращение к буферной памяти специального или удаленного модуля
Команды индикации	вывод знаков ASCII на выходы модуля или на светодиодный дисплей центрального процессора
Команды диагностики и устранения ошибок	проверка на наличие ошибок, фиксация состояния, семплирование (выборочная трассировка), контроль программы (трассировка)
Команды для обработки строковых величин	обработка строковых величин (ASCII-кодов)
Команды для особых функций	команды угловых функций, вычисление корней и степеней двоично-десятичных данных и чисел с плавающей запятой
Команды контроля данных	проверка входных данных нахождение в заданных диапазонах и сохранение проверенных данных
Команды переключения блоков регистров данных и файлов	переключение между блоками регистра файлов и файлами
Команды для часов	запись и считывание времени
Команды для периферийных устройств	вывод сообщений и ввод с клавиатуры на периферийных устройствах
Команды управления программой	команды для смены режимов выполнения программы
Прочие команды	сброс сторожевого таймера, установка и сброс флага переноса, выработка импульсов, непосредственное считывание и запись байтов, ввод с клавиатуры, сохранение и восстановление содержимого индексных регистров, запись данных в регистр EEPROM

## 7.1 Логические команды

С помощью логических команд программируются логические соединения, например, логическая сумма или логическое произведение.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе IEC
Логическое "И" (логическое умножение)	WAND	WAND_M, WAND_3_M
	WANDP	WANDP_M, WANDP_3_M
	DAND	DAND_M, DAND_3_M
	DANDP	DANDP_M, DANDP_3_M
	BKAND	BKAND_M
	BKANDP	BKANDP_M
Логическое "ИЛИ" (логическое сложение)	WOR	WOR_M, WOR_3_M
	WORP	WORP_M, WORP_3_M
	DOR	DOR_M, DOR_3_M
	DORP	DORP_M, DORP_3_M
	BKOR	BKOR_M
	BKORP	BKORP_M
Исключающее ИЛИ (XOR)	WXOR	WXOR_M, WXOR_3_M
	WXORP	WXORP_M, WXORP_3_M
	DXOR	DXOR_M, DXOR_3_M
	DXORP	DXORP_M, DXORP_3_M
	BKXOR	BKXOR_M
	BKXORP	BKXORP_M
Исключающее НЕ ИЛИ (XNR)	WXNR	WXNR_M, WXNR_3_M
	WXNRP	WXNRP_M, WXNRP_3_M
	DXNR	DXNR_M, DXNR_3_M
	DXNRP	DXNRP_M, DXNRP_3_M
	BKXNR	BKXNR_M
	BKXNRP	BKXNRP_M

**ПРИМЕЧАНИЕ** В редакторах по стандарту IEC используйте стандартные команды IEC.

Логические команды обрабатываются побитно в двоичной системе. При этом два состояния (0 и 1) логически увязываются друг с другом и результат этой логической связи выводится на целевой адрес.

В нижеследующей таблице истинности перечислены результаты различных логических соединений состояний 0 и 1. При этом А и В являются входными переменными, а Y – выходной переменной.

Логическая связь	Обработка	Логическая алгебра	Пример		
			А	В	Y
Функция "И"	Выход Y имеет состояние 1 только в том случае, если входы А и В одновременно имеют состояние 1.	$Y = A \times B$	0	0	0
			0	1	0
			1	0	0
			1	1	1
Функция "ИЛИ"	Выход Y имеет состояние 1, если по меньшей мере один из входов А или В имеет состояние 1.	$Y = A + B$	0	0	0
			0	1	1
			1	0	1
			1	1	1
Функция "исключающее ИЛИ"	Выход Y имеет состояние 1, если А и В неравны, и имеет состояние 0, если А и В равны.	$Y = \bar{A} \times B + A \times \bar{B}$	0	0	0
			0	1	1
			1	0	1
			1	1	0
Функция "исключающее НЕ ИЛИ"	Выход Y имеет состояние 1, если А и В равны, и имеет состояние 0, если А и В неравны.	$Y = (\bar{A} + B) (A + \bar{B})$	0	0	1
			0	1	0
			1	0	0
			1	1	1

7.1.1 WAND, WANDP, DAND, DANDP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

		Операнды																Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки			
		битовые						словные (16 бит)						Константы		Указатели							Уровень		
		X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V							K	H (16#)
<b>WAND</b>																									
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	5	●	●
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							1		
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					7		
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					1		
d1		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						1		
<b>DAND</b>																									
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K8	9	●	●
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					1		

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

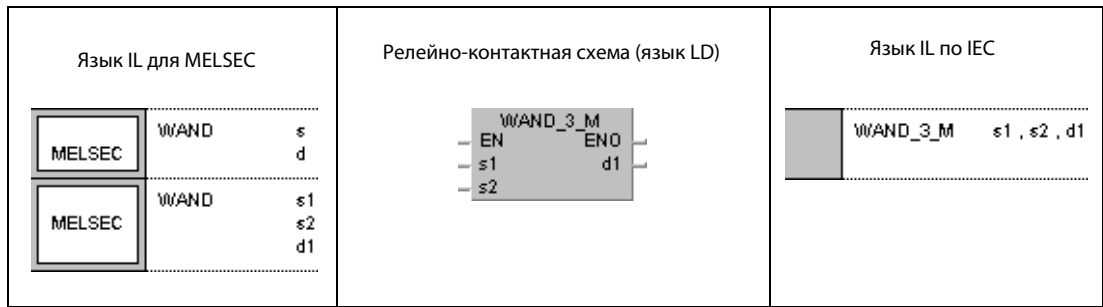
Операнды MELSEC Q

		Операнды								Флаг ошибки	Шагов	
		Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□V□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
		битовые	словные		битовые	словные						
<b>WAND</b>												
s	●	●	●	●	●	●	●	●	●	—	3	
d	●	●	●	●	●	●	●	—	—	—		
s1	●	●	●	●	●	●	●	●	●	—	4	
s2	●	●	●	●	●	●	●	●	—	—		
d1	●	●	●	●	●	●	●	—	—	—		
<b>DAND</b>												
s	●	●	●	●	●	●	●	●	●	—	4 <sup>1)</sup>	
d	●	●	●	●	●	●	●	—	—	—		
s1	●	●	●	●	●	●	●	●	●	—	4 <sup>2)</sup>	
s2	●	●	●	●	●	●	●	●	—	—		
d	●	●	●	●	●	●	●	—	—	—		

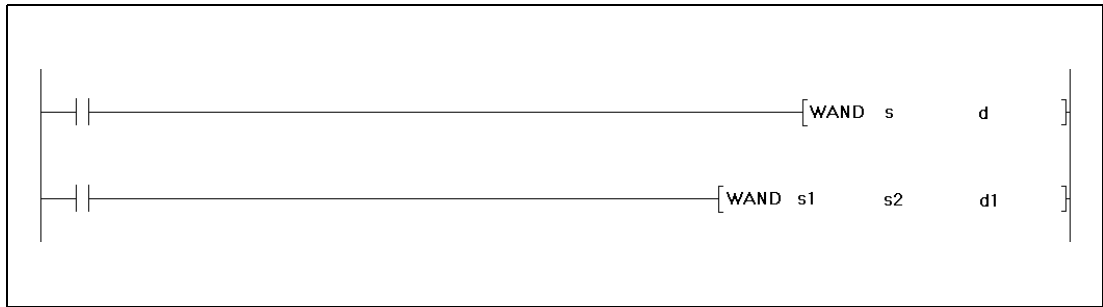
<sup>1</sup> Количество шагов программы зависит от типа центрального процессора и используемых операндов.  
 При использовании процессоров QnA: 4  
 В случае однопроцессорного модуля ЦПУ серии "Q": 3  
 В случае многопроцессорного ЦПУ серии "Q", внутренних словных операндов (кроме регистра файлов ZR) или констант: 6  
 В случае многопроцессорного ЦПУ серии "Q" и битовых операндов с адресом, кратным 16, имеющих обозначение битового блока K4 и не индексируемых: 6  
 При использовании многопроцессорного ЦПУ серии "Q" и иных операндов кроме вышеуказанных: 4

<sup>2</sup> Количество шагов программы зависит от типа центрального процессора и используемых операндов.  
 При использовании процессоров QnA: 4  
 В случае процессора "Q" и внутренних словных операндов (кроме регистра файлов ZR) или констант: 6  
 В случае процессора "Q" и битовых операндов с адресом, кратным 16, имеющих обозначение битового блока K4 и не индексируемых: 6  
 При использовании процессора Q и иных операндов кроме вышеуказанных: 4

**GX IEC Developer**



**GX Developer**



**Переменные**

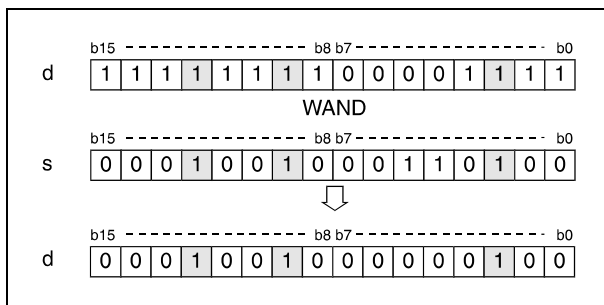
Операнд	Значение	Тип данных
s	Данные, в отношении которых выполняется операция логического умножения, или начальный адрес, начиная с которого эти данные сохранены.	BIN, 16/32 бита
d		
s1	Данные, в отношении которых выполняется операция логического умножения, или начальный адрес, начиная с которого эти данные сохранены.	
s2		
d1 (в случае DAND: d)	Начальный адрес, начиная с которого сохраняется результат.	

**Принцип действия****Логическое "И"****WAND 16-битные данные**

Логическое "И" (по-английски: AND) образует логическое произведение двух входных переменных.

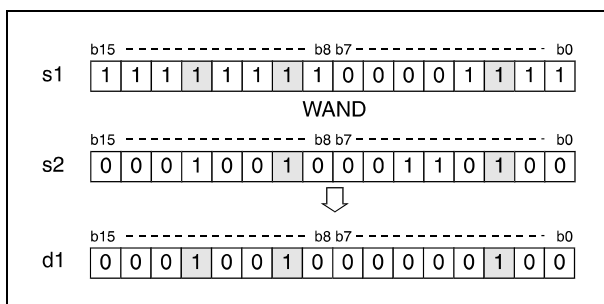
- 1-й вариант:

Побитно вычисляется логическое произведение 16-битных данных, указанных в s и d. Результат выводится в операнд, указанный в d.



- 2-й вариант:

Побитно вычисляется логическое произведение 16-битных данных, указанных в s1 и s2. Результат выводится в операнд, указанный в d1.

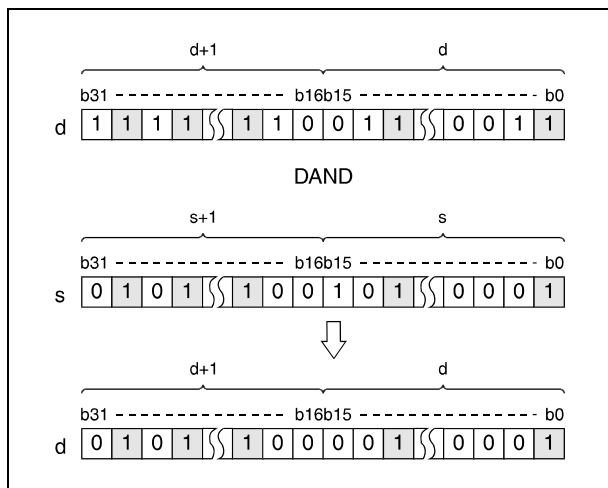


Биты сверх длины блока устанавливаются на 0. Например, если установлена длина блока K2, то старшие 8 битов (с b8 по b15) интерпретируются как биты с состоянием 0.

**DAND 32-битные данные**

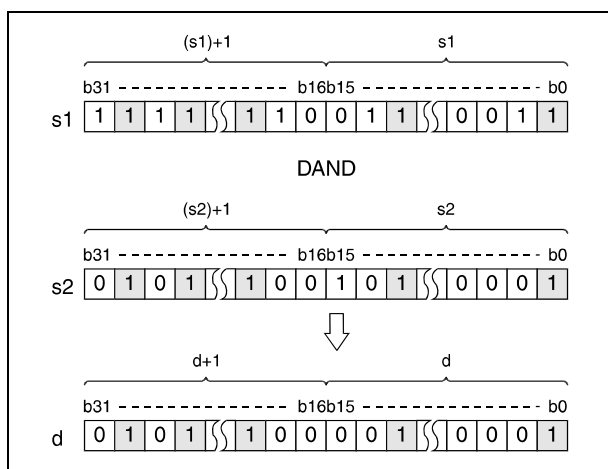
● 1-й вариант:

Побитно вычисляется логическое произведение 32-битных данных, указанных в s и d. Результат выводится в операнд, указанный в d.



● 2-й вариант (серия QnA/"System Q"):

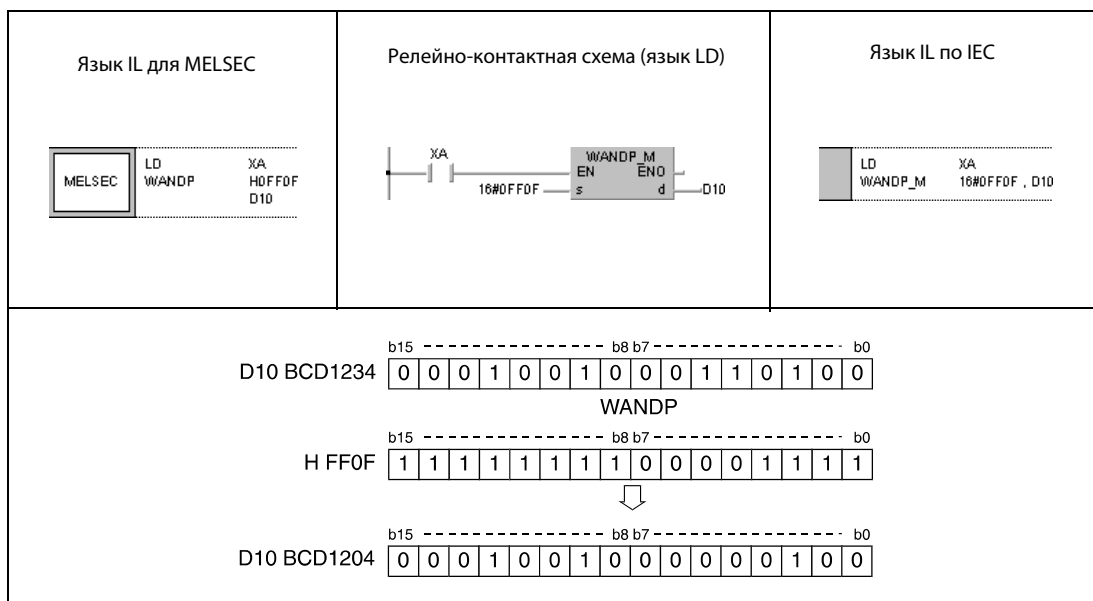
Побитно вычисляется логическое произведение 32-битных данных, указанных в s1 и s2. Результат выводится в операнд, указанный в d.



После выполнения соединения все биты, находящиеся вне области блока, устанавливаются на 0.

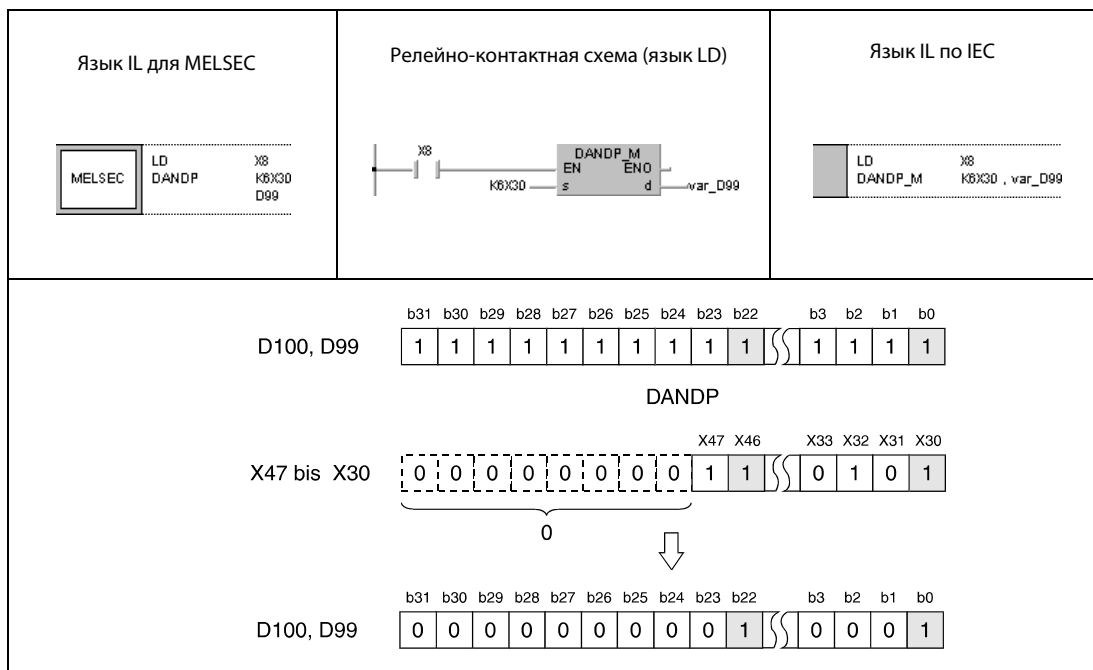
**Пример 1** WANDP (s, d)

Следующая программа при положительном (нарастающем) фронте сигнала XA устанавливает разряд десятков (b5–b7) указанного в D10 двоично-десятичного значения на 0 и снова сохраняет результат в D10.



**Пример 2** DANDP (s, d)

Следующая программа при положительном фронте сигнала X8 образует логическое произведение 32-битного значения данных в D99 и D100 и 24-битного значения данных X30...X47 и снова сохраняет результат в D99 и D100.

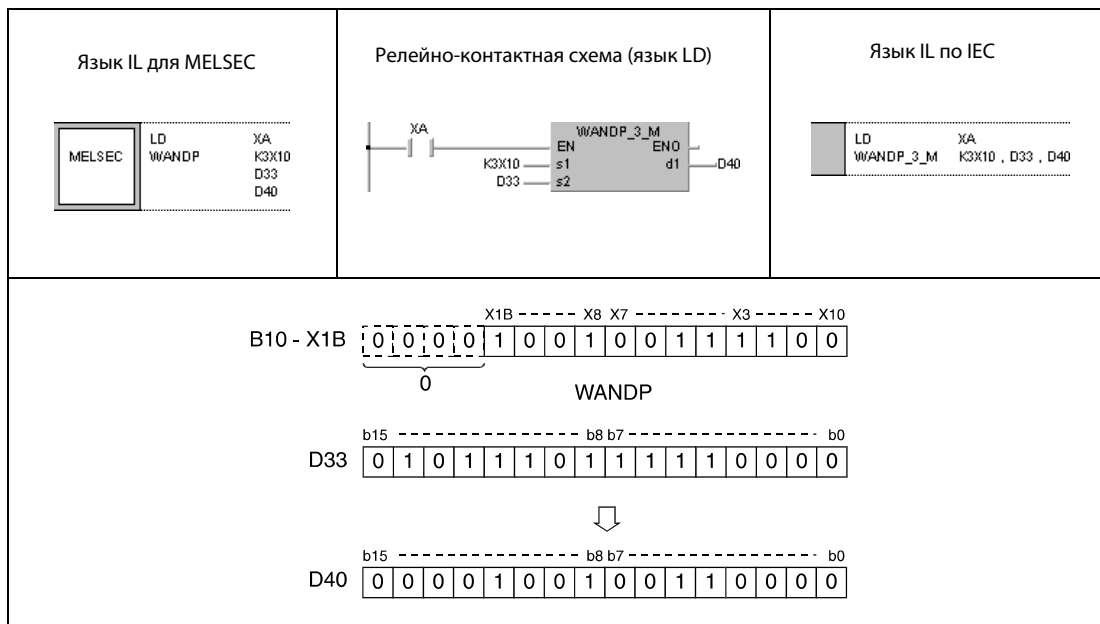


<sup>1</sup> Эти биты считываются в качестве 0.



**Пример 3** WANDP (s1, s2, d1)

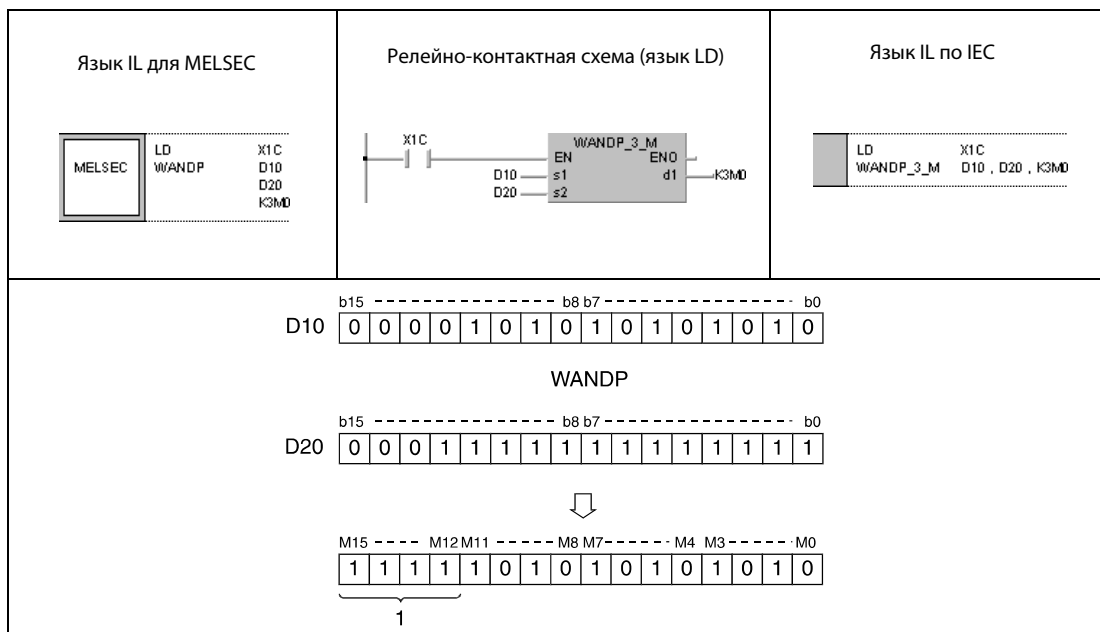
Следующая программа при положительном фронте сигнала XA образует логическое произведение данных X10...X1B и данных в D33 и сохраняет результат по адресу D40.



<sup>1</sup> Эти биты считываются в качестве 0.

**Пример 4** WANDP (s1, s2, d1)

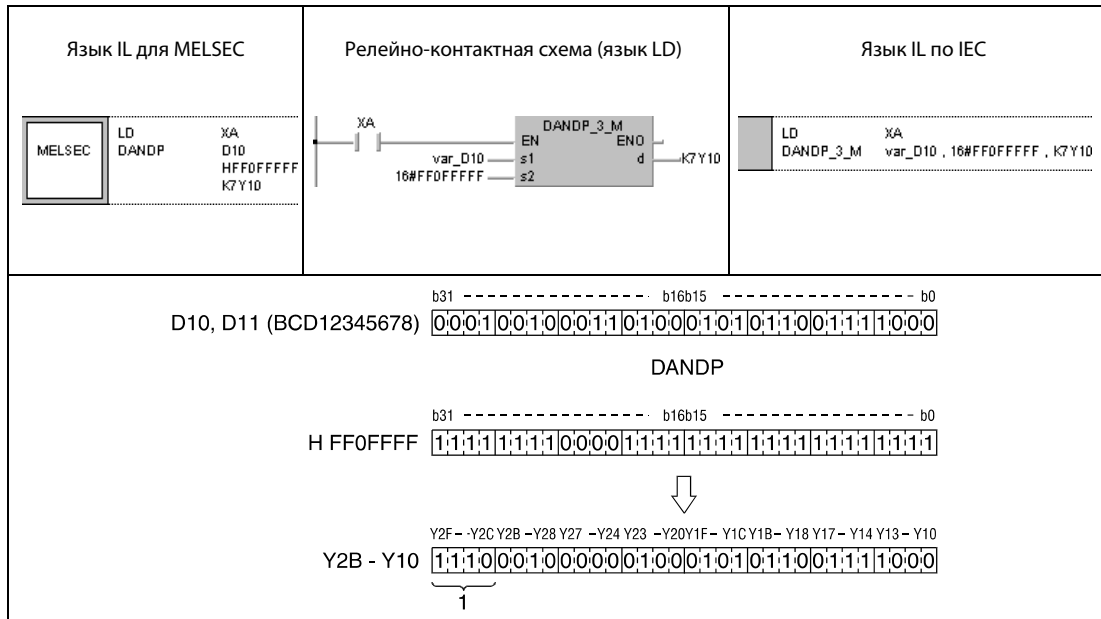
Следующая программа при положительном фронте сигнала X1C образует логическое произведение данных в D10 и D20 и сохраняет результат в маркерах с M0 по M11.



<sup>1</sup> Эти биты не изменяют свое состояние.

**Пример 5** DANDP (s1, s2, d)

Следующая программа при положительном фронте сигнала XA устанавливает разряд сотен тысяч указанного в D10 и D11 двоично-десятичного значения на 0 и выводит результат на выходы с Y10 по Y2B.



<sup>1</sup> Эти биты не изменяют свое состояние.

**ПРИМЕЧАНИЕ** Без объявления переменной в заголовке программного компонента (POU) примеры программ 2 и 5 не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

**7.1.2 BKAND, BKANDP**

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	SMO	5	
s2	—	●	●	—	—	—	—	●			
d	—	●	●	—	—	—	—	—			
n	●	●	●	●	●	●	●	●			

GX IEC Developer



GX Developer



Переменные

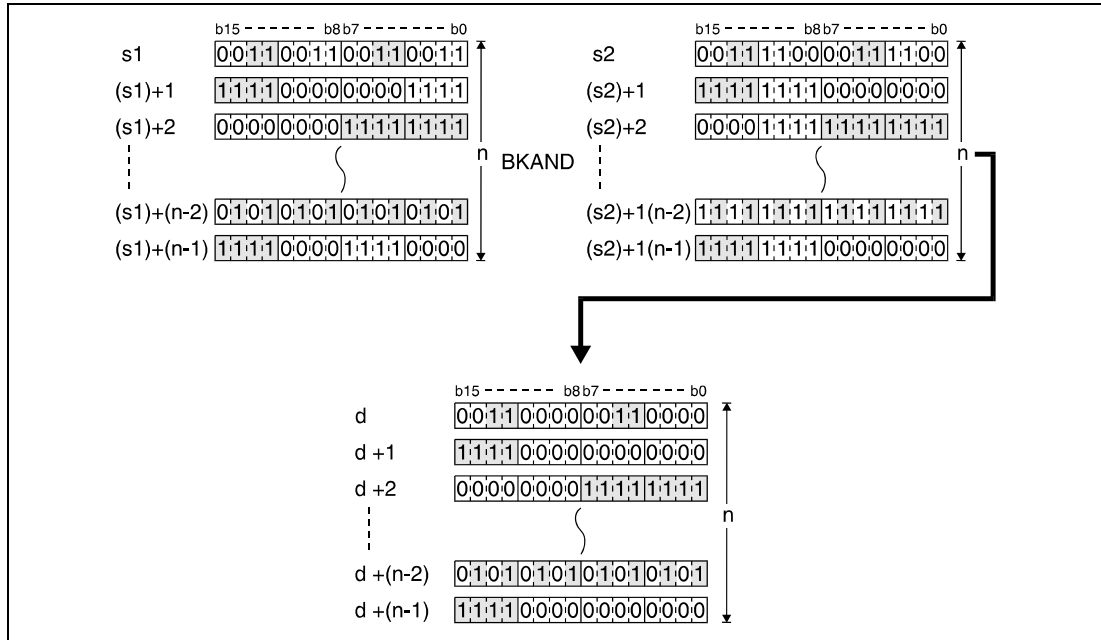
Операнд	Значение	Тип данных
s1	Первый адрес операнда, в котором сохранены данные для операции.	BIN, 16 бит
s2	Первый адрес данных для операции или первый адрес операнда, в котором эти данные сохранены.	
d	Первый адрес операнда, в котором сохраняется результат.	
n	Количество логически умножаемых блоков данных	

**Принцип действия**

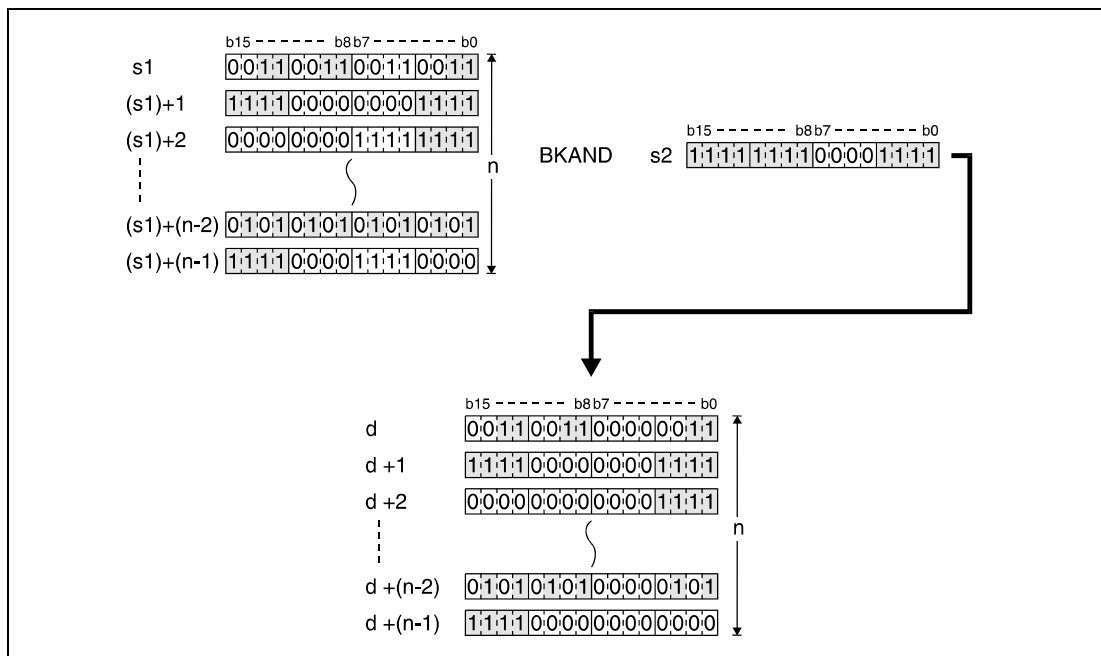
**Логическое умножение 16-битных блоков данных**

**ВКАНД Поблочное логическое умножение**

Команда ВКАНД образует логическое произведение n 16-битных блоков, начиная с s1, и n-16-битных блоков, начиная с s2. Соответствующий 16-битный блок результата сохраняется начиная с операнда, указанного в d. Количество блоков, с которыми выполняется эта операция, указано в n.



Хранящаяся в s2 константа должна находиться в диапазоне между -32768 и 32767.



**Источники ошибок**

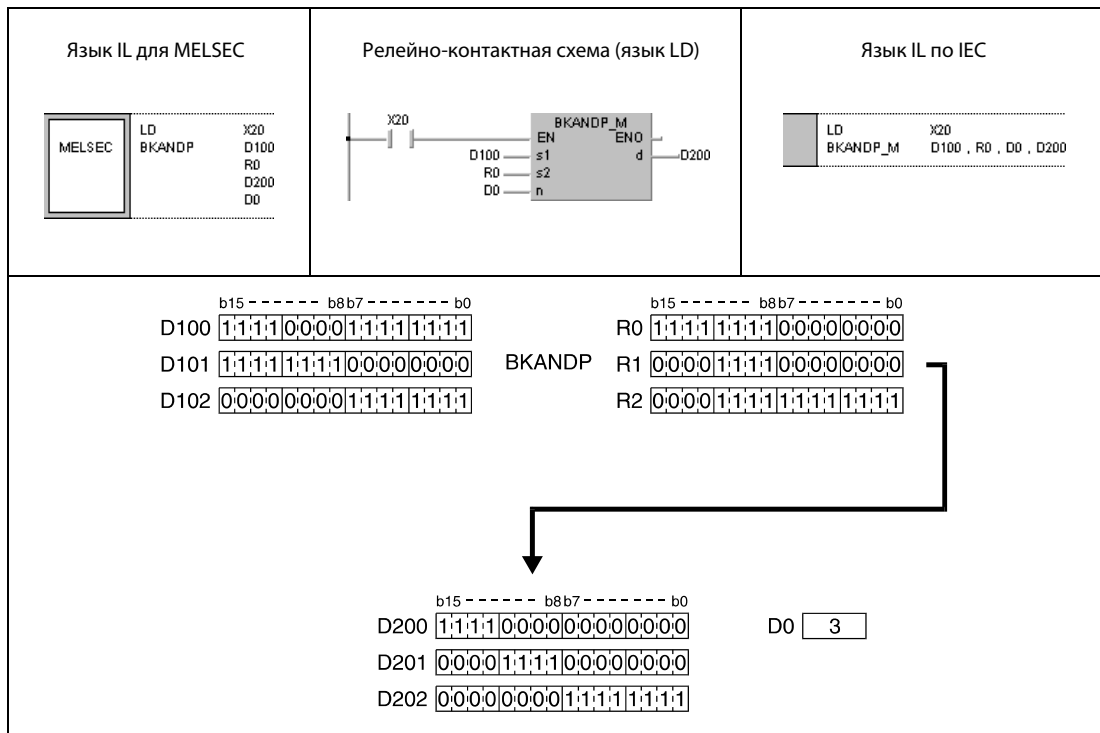
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в n количество блоков s1, s2 или d находится вне области операнда, предусмотренной для сохранения (код ошибки 4101).
- Предусмотренные для сохранения области s1, s2 или d перекрываются (код ошибки 4101).

**Пример**

**BKANDP**

Следующая программа при положительном фронте X20 образует логическое произведение данных в регистрах D100...D102 и данных в регистрах R0...R2. Результат сохраняется в регистрах с D200 по D202. Количество участвующих в операции 16-битных блоков данных (3) указано в D0.



7.1.3 WOR, WOPR, DOR, DORP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

Операнды																		Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011			
битовые						словные (16 бит)						Константы		Указатели		Уровень									
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V		K						H (16#)	P	I
<b>WOR</b>																									
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					K1 ↓ K4	5	●	●
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					7				
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					7				
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					7				
d1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					7				
<b>DOR</b>																									
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					K1 ↓ K8	9	●	●
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					9				

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

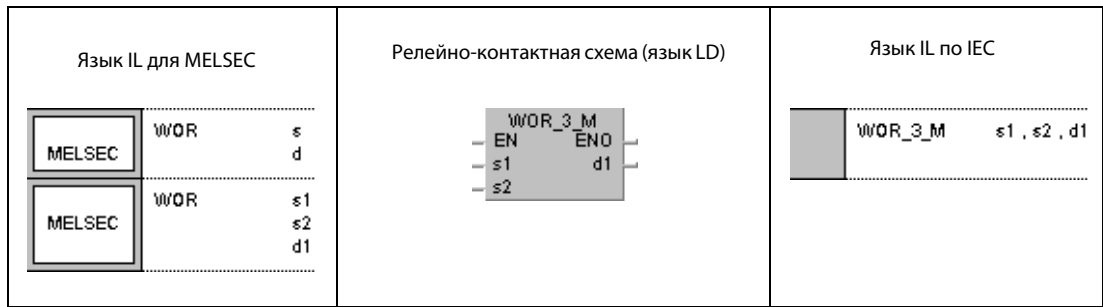
Операнды MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 неопоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы K, H (16#)	Иные U			
битовые	словные		битовые	словные							
<b>WOR</b>											
s	●	●	●	●	●	●	●	●	—	—	3
d	●	●	●	●	●	●	●	—	—		
s1	●	●	●	●	●	●	●	●	—	4	
s2	●	●	●	●	●	●	●	●	—		
d1	●	●	●	●	●	●	●	—	—		
<b>DOR</b>											
s	●	●	●	●	●	●	●	●	—	4 <sup>1)</sup>	
d	●	●	●	●	●	●	●	—	—		
s1	●	●	●	●	●	●	●	●	—	4 <sup>2)</sup>	
s2	●	●	●	●	●	●	●	●	—		
d	●	●	●	●	●	●	●	—	—		

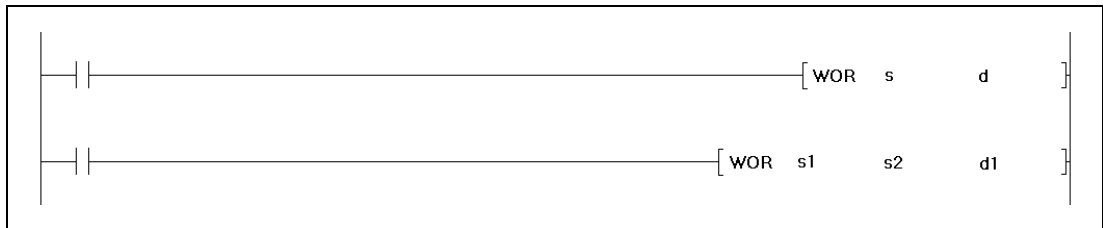
<sup>1</sup> Количество шагов программы зависит от типа центрального процессора и используемых операндов.  
 При использовании процессоров QnA: 4  
 В случае однопроцессорного модуля ЦП серии "Q": 3  
 В случае многопроцессорного ЦПУ серии "Q", внутренних словных операндов (кроме регистра файлов ZR) или констант: 6  
 В случае многопроцессорного ЦПУ серии "Q" и битовых операндов с адресом, кратным 16, имеющих обозначение битового блока K4 и не индексируемых: 6  
 При использовании многопроцессорного ЦПУ серии "Q" и иных операндов кроме вышеуказанных: 4

<sup>2</sup> Количество шагов программы зависит от типа центрального процессора и используемых операндов.  
 При использовании процессоров QnA: 4  
 В случае процессора "Q" и внутренних словных операндов (кроме регистра файлов ZR) или констант: 6  
 В случае процессора "Q" и битовых операндов с адресом, кратным 16, имеющих обозначение битового блока K4 и не индексируемых: 6  
 При использовании процессора Q и иных операндов кроме вышеуказанных: 4

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Логически суммируемые данные или начальный адрес операнда, в котором эти данные сохранены.	BIN, 16/32 бита
d		
s1	Логически суммируемые данные или начальный адрес операнда, в котором эти данные сохранены.	
s2		
d1 (в случае DOR: d)	Начальный адрес операнда, в котором сохраняется результат.	

**Принцип действия**

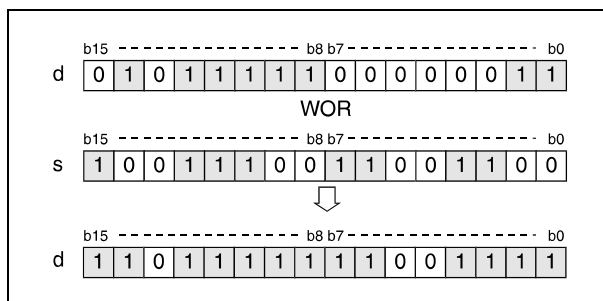
**Логическое "ИЛИ"**

**WOR 16-битные данные**

Операция "логическое ИЛИ" (по-английски: OR) образует логическую сумму двух входных переменных.

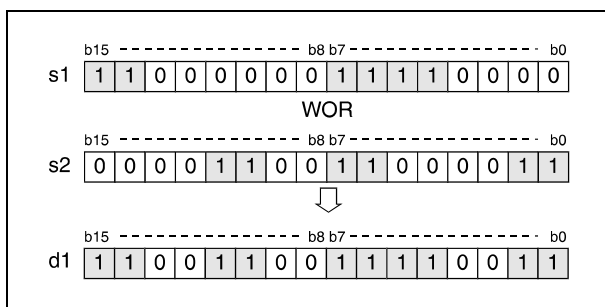
- 1-й вариант:

Побитно суммируются 16-битные данные, указанные в s и d. Результат выводится в операнд, указанный в d.



● 2-й вариант:

Побитно суммируются 16-битные данные, указанные в s1 и s2. Результат выводится в операнд, указанный в d1.



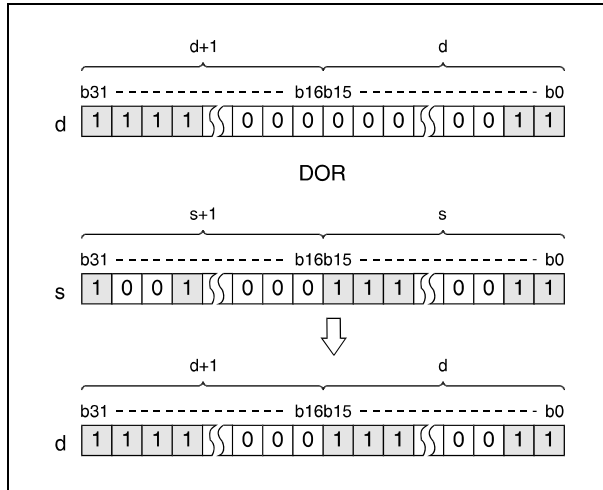
Биты сверх длины блока устанавливаются на 0. Например, если установлена длина блока K2, то старшие 8 битов (с b8 по b15) интерпретируются как биты с состоянием 0.



**DOR 32-битные данные**

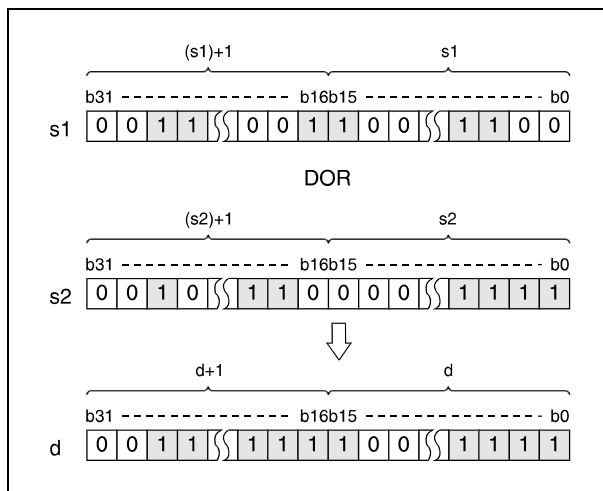
● 1-й вариант:

Побитно суммируются 32-битные данные, указанные в s и d. Результат выводится в операнд, указанный в d.



● 2-й вариант (серия QnA/"System Q"):

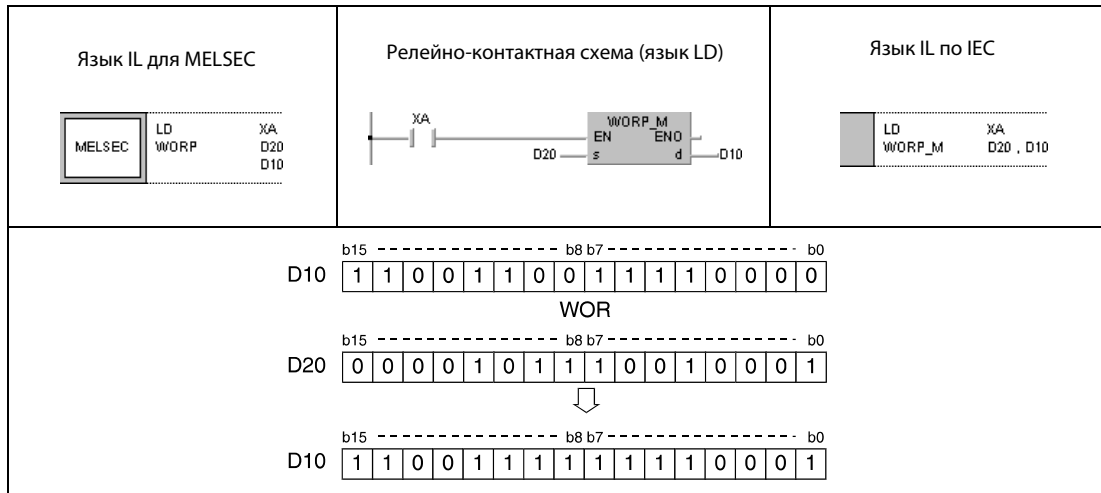
Побитно суммируются 32-битные данные, указанные в s1 и s2. Результат выводится в операнд, указанный в d.



После выполнения соединения все биты, находящиеся вне области блока, устанавливаются на 0.

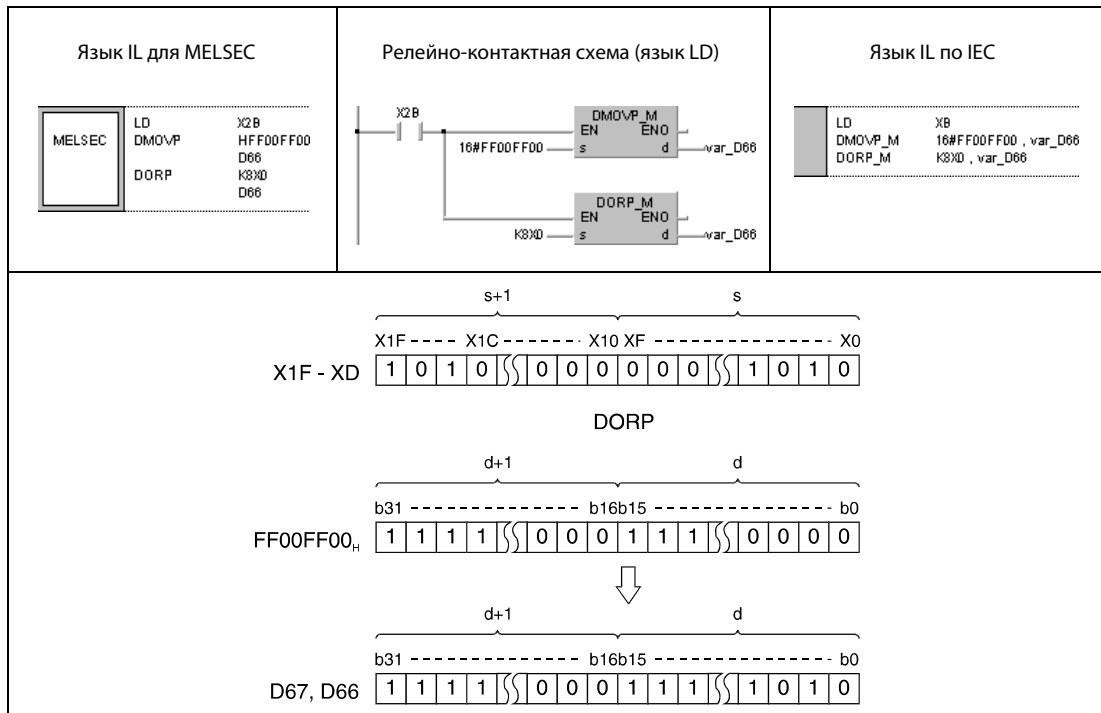
**Пример 1** WORP (s, d)

Следующая программа при положительном фронте сигнала XA суммирует данные из D10 с данными из D20. Результат сохраняется в D10.



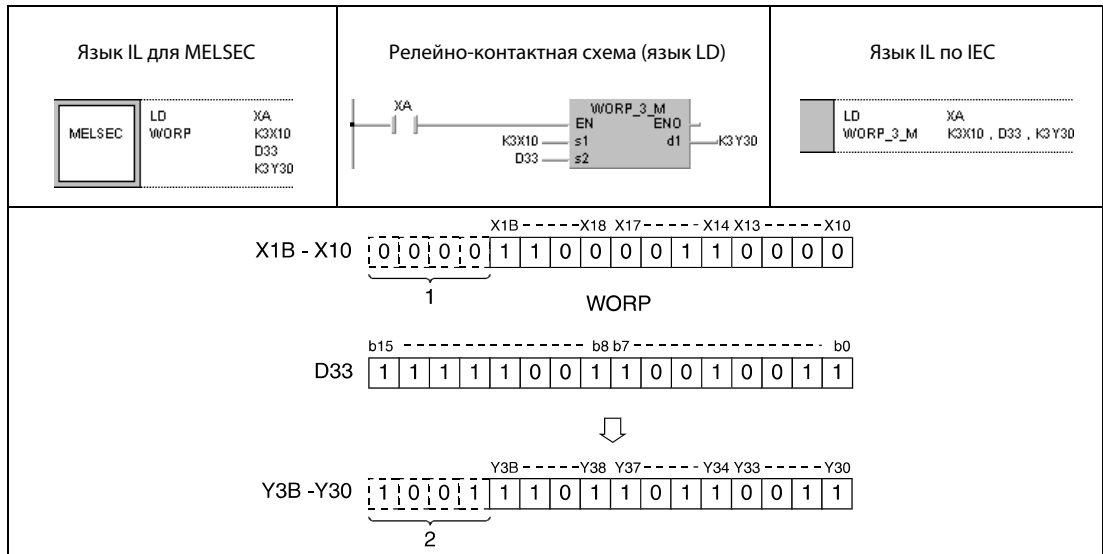
**Пример 2** DORP (s, d)

Следующая программа при положительном фронте сигнала X2B суммирует данные входов X0...X1F с шестнадцатеричным значением FF00FF00. Результат сохраняется в D66 и D67.



**Пример 3**      WORP (s1, s2, d1)

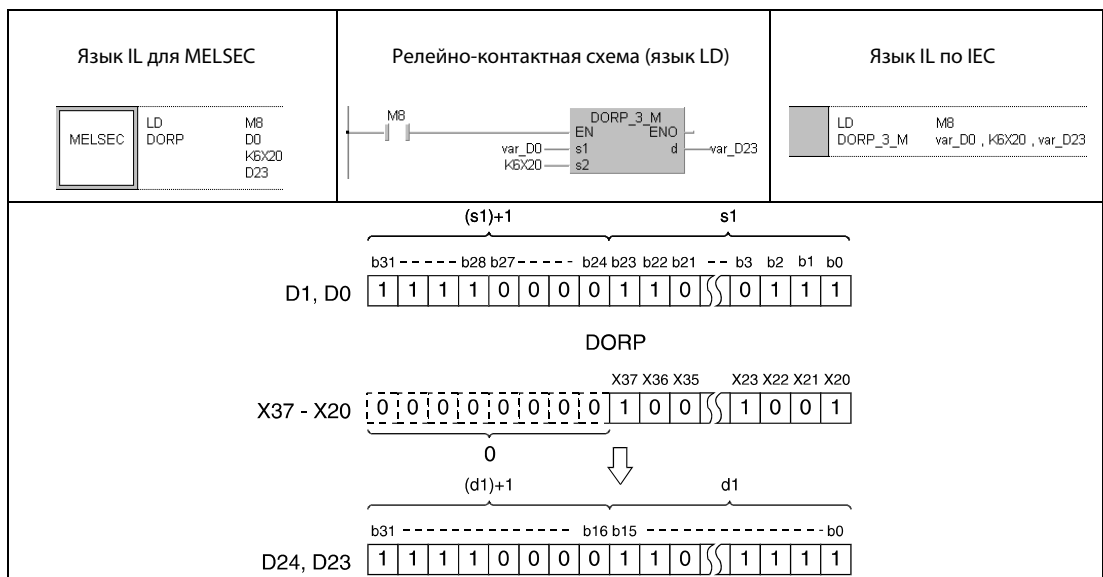
Следующая программа при положительном фронте сигнала XA суммирует данные входов X10...X1В с данными из D33. Результат выводится на выходы Y30...Y3В.



- <sup>1</sup> Эти биты считаются в качестве 0.
- <sup>2</sup> Эти биты не изменяют свое состояние.

**Пример 4**      DORP (s1, s2, d)

Следующая программа при положительном фронте M8 суммирует 32-битные данные из D0 и D1 с 24-битными данными входов X20...X37. Результат сохраняется в D23 и D24.



**ПРИМЕЧАНИЕ**      Без объявления переменной в заголовке программного компонента (POU) примеры программ 2 и 4 не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

7.1.4 БКОР, ВКОРР

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

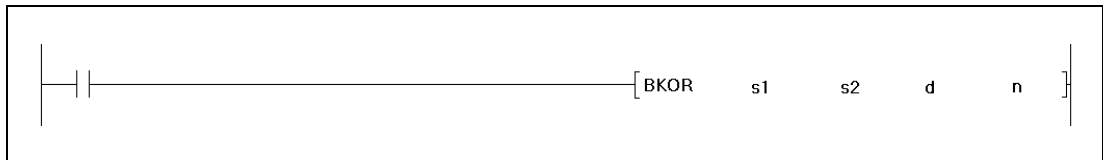
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	SMO	5	
s2	—	●	●	—	—	—	—	●			
d	—	●	●	—	—	—	—	—			
n	●	●	●	●	●	●	●	—			

GX IEC Developer



GX Developer



Переменные

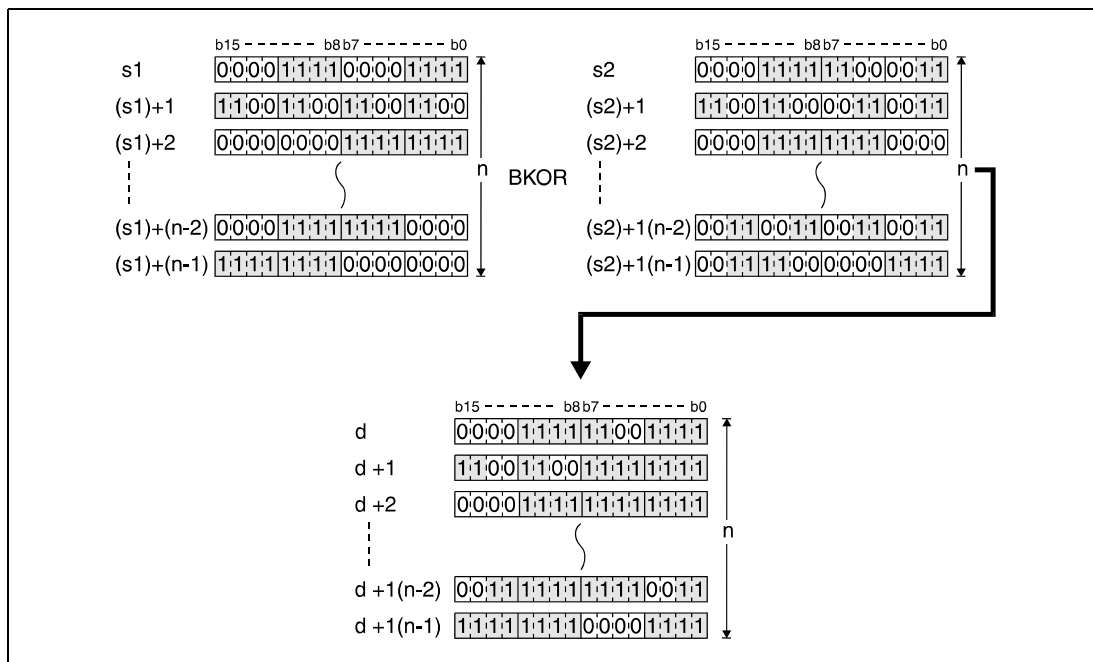
Операнд	Значение	Тип данных
s1	Первый адрес операнда, в котором сохранены данные для операции.	BIN, 16 бит
s2	Первый адрес данных для операции или первый адрес операнда, в котором эти данные сохранены.	
d	Первый адрес операнда, в котором сохраняется результат.	
n	Количество логически суммируемых блоков данных	

**Принцип действия**

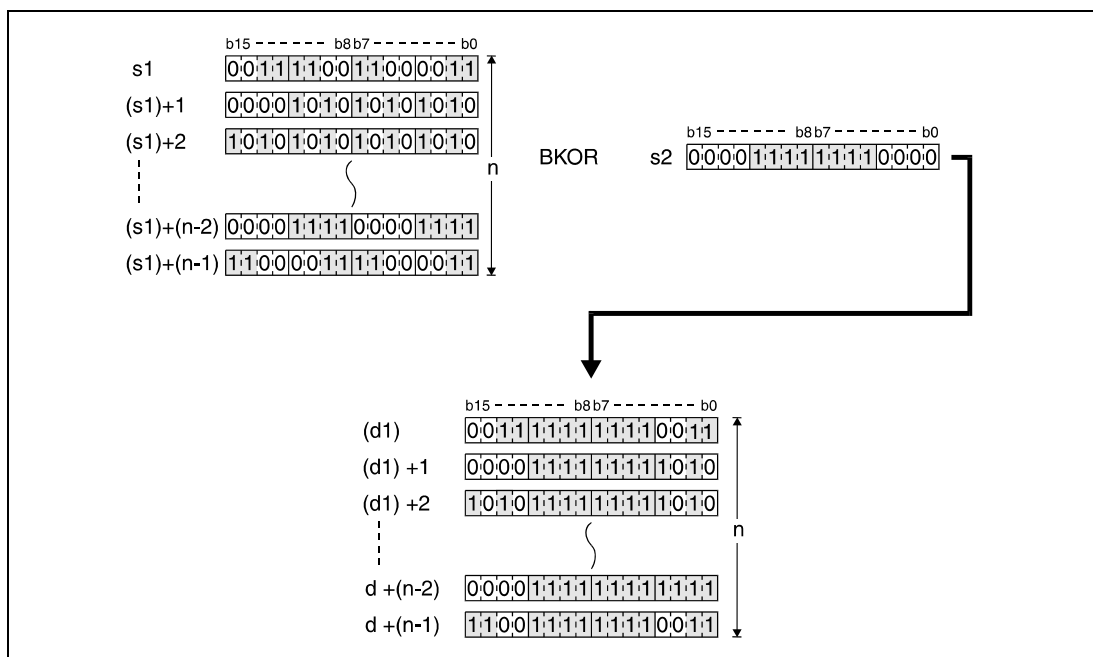
**Логическое сложение 16-битных блоков данных**

**ВКОР Поблочное логическое сложение**

Команда ВКОР образует логическую сумму каждого из n 16-битных блоков, начиная с s1, и соответствующего 16-битного блока, начиная с s2. Соответствующий 16-битный блок результата сохраняется, начиная с операнда, указанного в d. Количество блоков, с которыми выполняется эта операция, указано в n.



Хранящаяся в s2 константа должна находиться в диапазоне между -32768 и 32767.



**Источники ошибок**

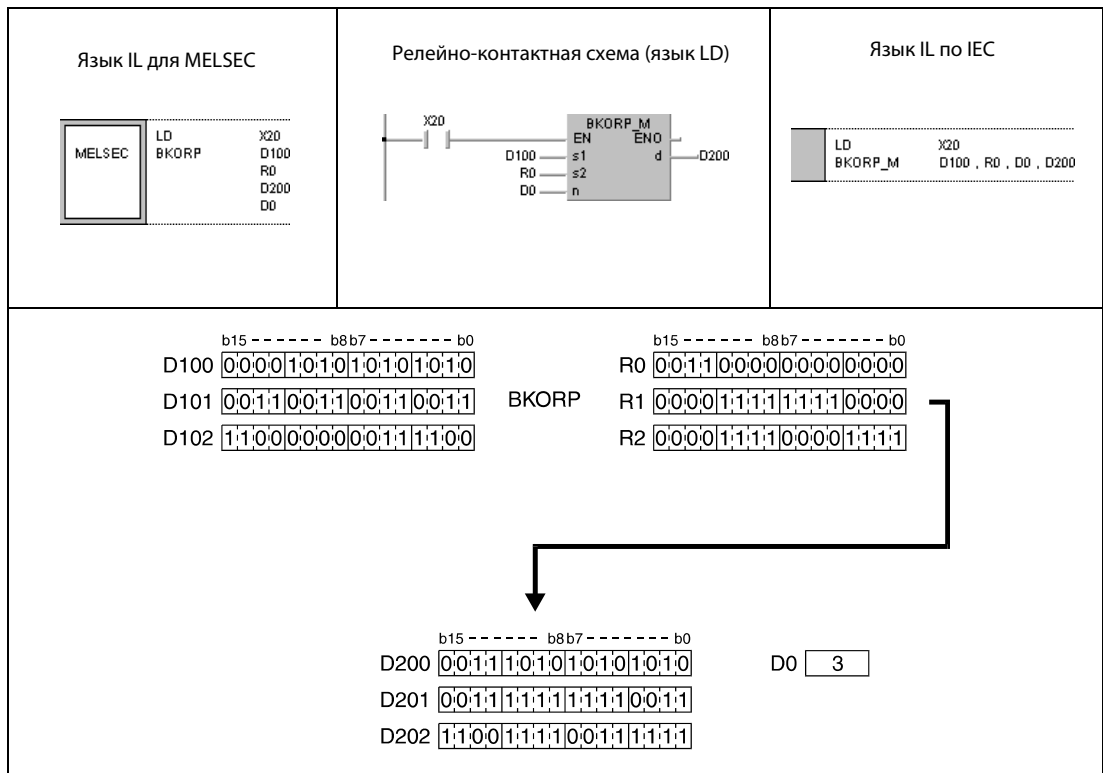
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в n количество блоков s1, s2 или d находится вне области операнда, предусмотренной для сохранения (код ошибки 4101).
- Предусмотренные для сохранения области s1, s2 или d перекрываются (код ошибки 4101).

**Пример**

**ВКОРР**

Следующая программа при положительном фронте X20 образует логическую сумму данных в регистрах D100...D102 и данных в регистрах R0...R2. Результат сохраняется в регистрах с D200 по D202. Количество участвующих в операции 16-битных блоков данных (3) указано в D0.



7.1.5 WXOR, WXORP, DXOR, DXORP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

		Операнды																Длина блока	Шагов	Индекс	Флаг переноса		Флаг ошибки							
		битовые						словные (16 бит)						Константы		Указатели					Уровень		M9012	M9010 M9011						
		X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V				K	H (16#)	P	I	N					
<b>WXOR</b>																														
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					K1 ↓ K4	5	1					●	●
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							7	1						
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						9	1						
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						K8	1						
d1		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●														
<b>DXOR</b>																														
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					K8	1						
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●														

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

Операнды MELSEC Q

		Операнды										Флаг ошибки	Шагов
		Внутренние операнды (систем., пользов.)		Регистр файлов	MELSECNET/10 неопр. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные			
		битовые	словные		битовые	словные							
<b>WXOR</b>													
s	●	●	●	●	●	●	●	●	●	●	—	—	3
d	●	●	●	●	●	●	●	●	●	—	—	—	
s1	●	●	●	●	●	●	●	●	●	●	—	—	4
s2	●	●	●	●	●	●	●	●	●	●	—	—	
d1	●	●	●	●	●	●	●	●	●	—	—	—	
<b>DXOR</b>													
	●	●	●	●	●	●	●	●	●	●	—	—	4 <sup>1)</sup>
	●	●	●	●	●	●	●	●	●	—	—	—	
s1	●	●	●	●	●	●	●	●	●	●	—	—	4 <sup>2)</sup>
s2	●	●	●	●	●	●	●	●	●	●	—	—	
d	●	●	●	●	●	●	●	●	●	—	—	—	

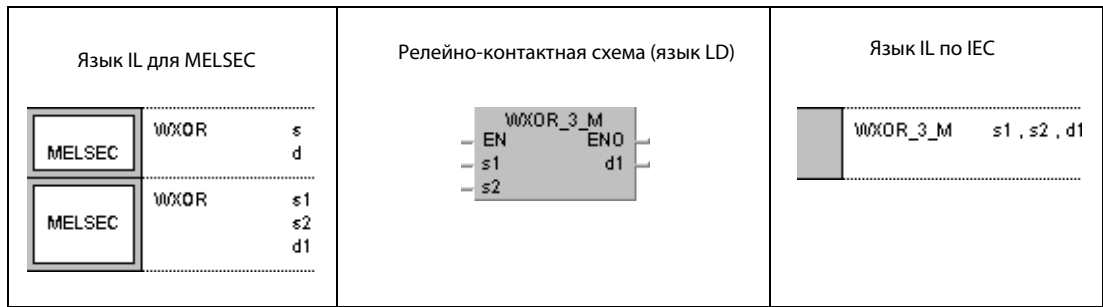
<sup>1</sup> Количество шагов программы зависит от типа центрального процессора и используемых операндов.

- При использовании процессоров QnA: 4
- В случае однопроцессорного модуля ЦП серии "Q": 3
- В случае многопроцессорного ЦПУ серии "Q", внутренних словных операндов (кроме регистра файлов ZR) или констант: 6
- В случае многопроцессорного ЦПУ серии "Q" и битовых операндов с адресом, кратным 16, имеющих обозначение битового блока K4 и не индексируемых: 6
- При использовании многопроцессорного ЦПУ серии "Q" и иных операндов кроме вышеуказанных: 4

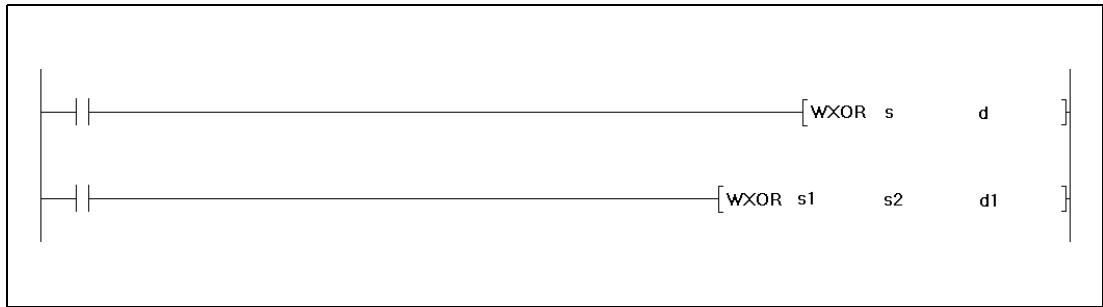
<sup>2</sup> Количество шагов программы зависит от типа центрального процессора и используемых операндов.

- При использовании процессоров QnA: 4
- В случае процессора "Q" и внутренних словных операндов (кроме регистра файлов ZR) или констант: 6
- В случае процессора "Q" и битовых операндов с адресом, кратным 16, имеющих обозначение битового блока K4 и не индексируемых: 6
- При использовании процессора Q и иных операндов кроме вышеуказанных: 4

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Данные, в отношении которых выполняется операция "исключающее ИЛИ", или первый адрес операнда, в котором сохранены эти данные.	BIN, 16/32 бита
d		
s1	Данные, в отношении которых выполняется операция "исключающее ИЛИ", или первый адрес операнда, в котором сохранены эти данные.	
s2		
d1 (в случае DXOR: d)	Первый адрес операнда, в котором сохраняется результат.	

**Принцип действия**

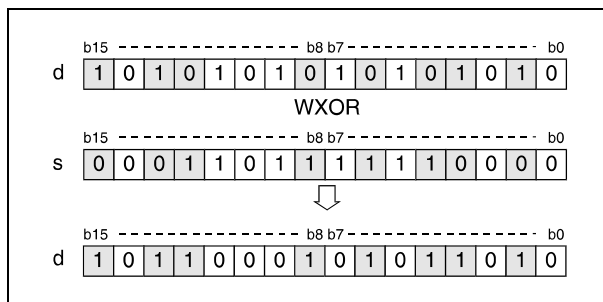
**Логическая операция "исключающее ИЛИ"**

**WXOR 16-битные данные**

Операция "исключающее ИЛИ" (по-английски exclusive OR) образует логическую сумму из произведения двух вводных переменных ( $Y = (\bar{A} \times B) + (A \times \bar{B})$ ).

- 1-й вариант:

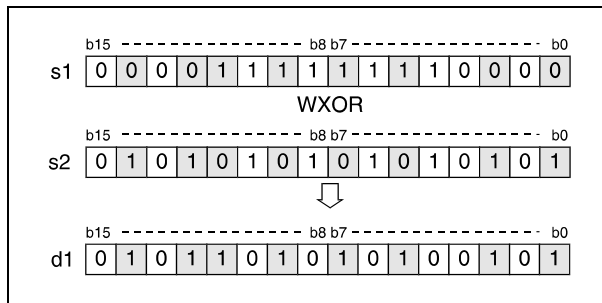
Логическая операция "исключающее ИЛИ" выполняется для 16-битных данных, указанных в s и d. Результат выводится в операнд, указанный в d.





## ● 2-й вариант:

Логическая операция "исключающее ИЛИ" выполняется для 16-битных данных, указанных в s1 и s2. Результат выводится в операнд, указанный в d1.

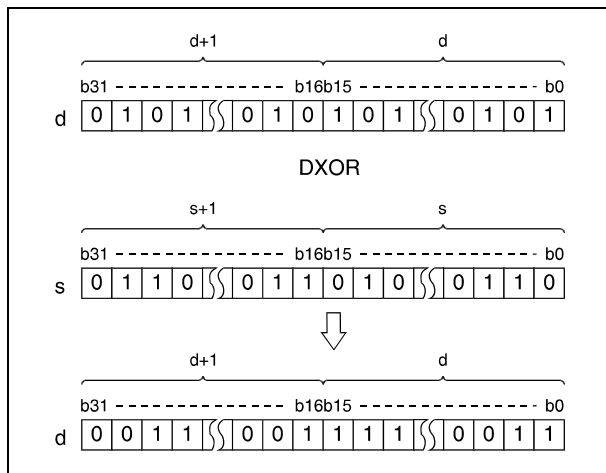


Биты сверх длины блока устанавливаются на 0. Например, если установлена длина блока K2, то старшие 8 битов (с b8 по b15) интерпретируются как биты с состоянием 0.

**DXOR 32-битные данные**

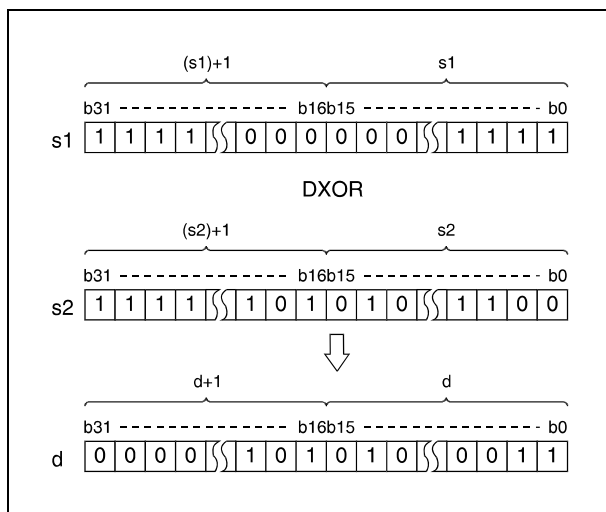
● 1-й вариант:

Логическая операция "исключающее ИЛИ" выполняется для 32-битных данных, указанных в s и d. Результат выводится в операнд, указанный в d.



● 2-й вариант (серия QnA/"System Q"):

Логическая операция "исключающее ИЛИ" выполняется для 32-битных данных, указанных в s1 и s2. Результат выводится в операнд, указанный в d.

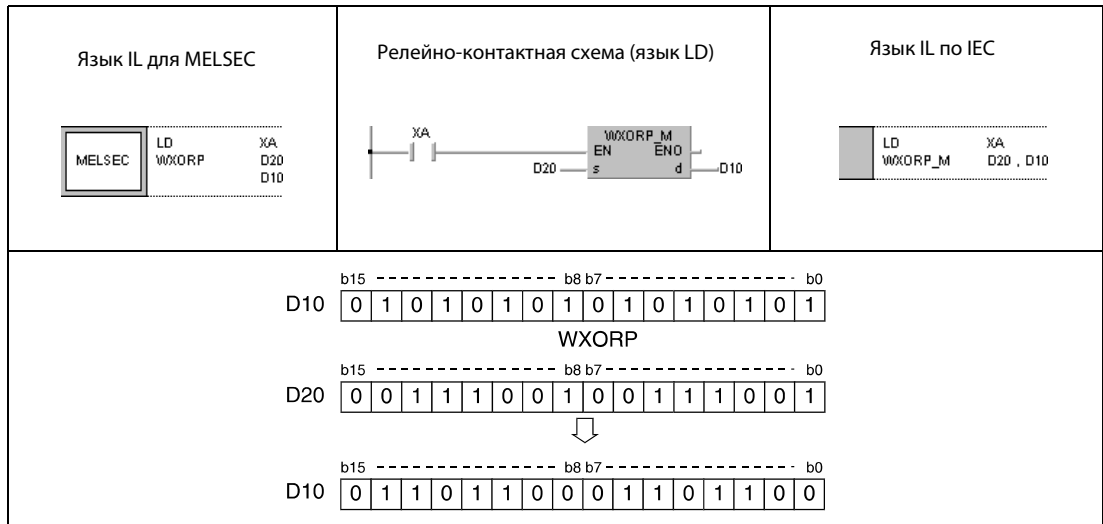


После выполнения соединения все биты, находящиеся вне области блока, устанавливаются на 0.

**ПРИМЕЧАНИЕ** Если применяется вариант 1 (s, d) и не применяется индексная адресация, то никакие источники ошибок при использовании команд WXOR, WXORP, DXOR и DXORP не известны.

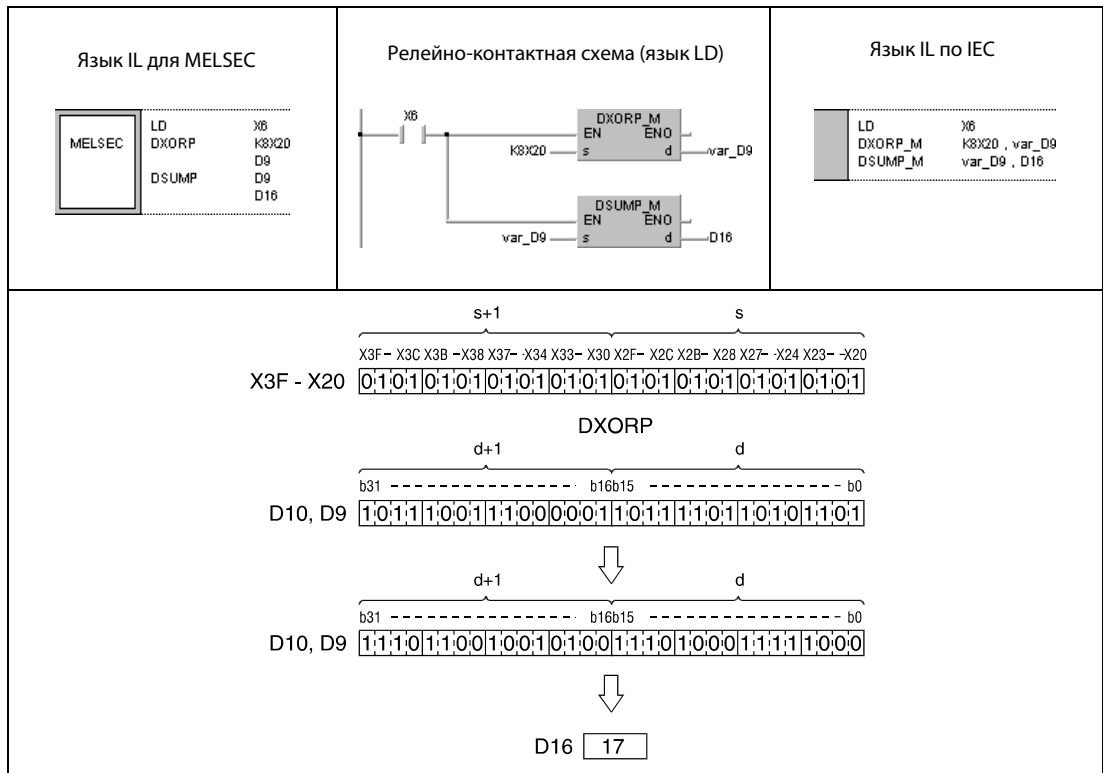
**Пример 1** WXORP (s, d)

Следующая программа при положительном фронте сигнала XA соединяет данные из D10 с данными из D20 и снова сохраняет результат в D10.



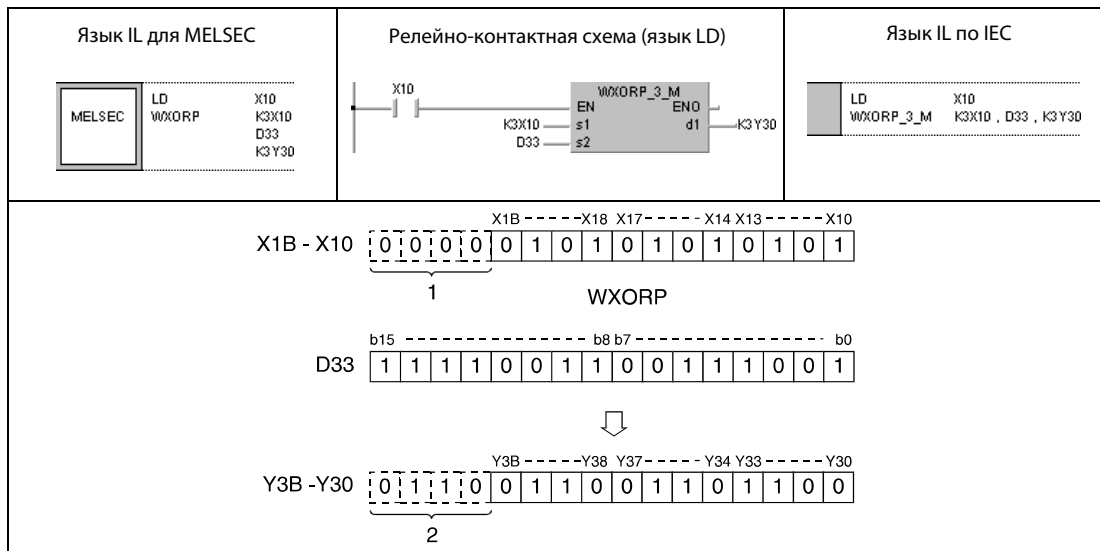
**Пример 2** DXORP (s, d)

Следующая программа сравнивает 32-битное значение входов X20...X3F с набором битов в регистре данных D9 и D10. Результат снова сохраняется в D9 и D10. Количество установленных битов в D9 и D10 сохраняется в D16. Операция "исключающее ИЛИ" выполняется при положительном фронте сигнала X6.



**Пример 3** WXORP (s1, s2, d1)

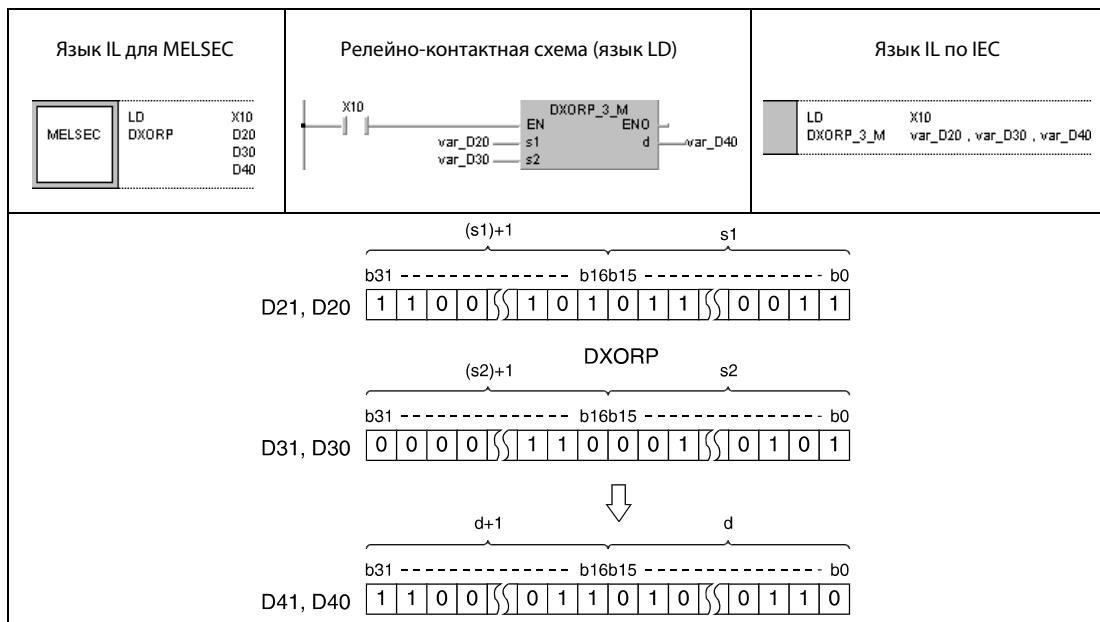
Следующая программа при положительном фронте сигнала X10 выполняет операцию "исключающее ИЛИ" для входных данных X10...X1B и данных в D33. Результат сохраняется в D33 и выводится на выходы Y30...Y3B.



- <sup>1</sup> Эти биты считываются в качестве 0.
- <sup>2</sup> Эти биты не изменяют свое состояние.

**Пример 4** DXORP (s1, s2, d)

Следующая программа при положительном фронте сигнала X10 выполняет операцию "исключающее ИЛИ" для данных в D20...D21 и данных в D30...D31. Результат сохраняется в D40 и D41.



**ПРИМЕЧАНИЕ** Без объявления переменной в заголовке программного компонента (POU) примеры программ 2 и 4 не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

**7.1.6 BKXOR, BKXORP**

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

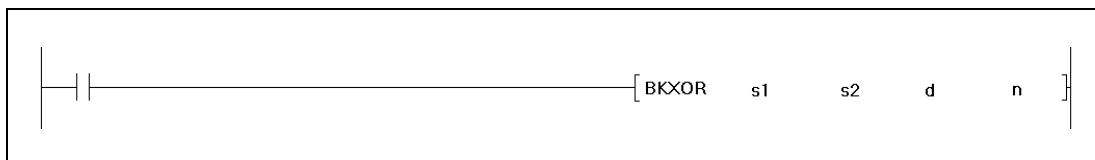
Операнды  
MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	—	SM0	5
s2	—	●	●	—	—	—	—	●	—		
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer



GX Developer



Переменные

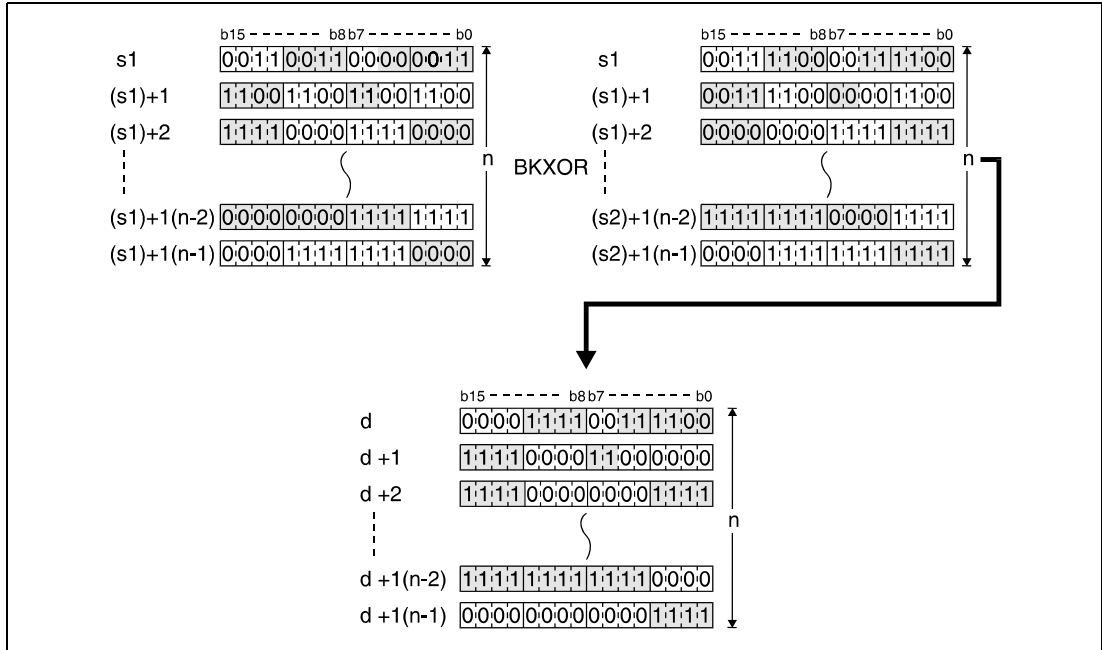
Операнд	Значение	Тип данных
s1	Первый адрес операнда, в котором сохранены данные для операции.	BIN, 16 бит
s2	Первый адрес данных или первый адрес операнда, в котором сохранены обрабатываемые данные.	
d	Первый адрес операнда, в котором сохраняется результат.	
n	Количество блоков данных, для которых выполняется операция "исключающее ИЛИ".	

**Принцип действия**

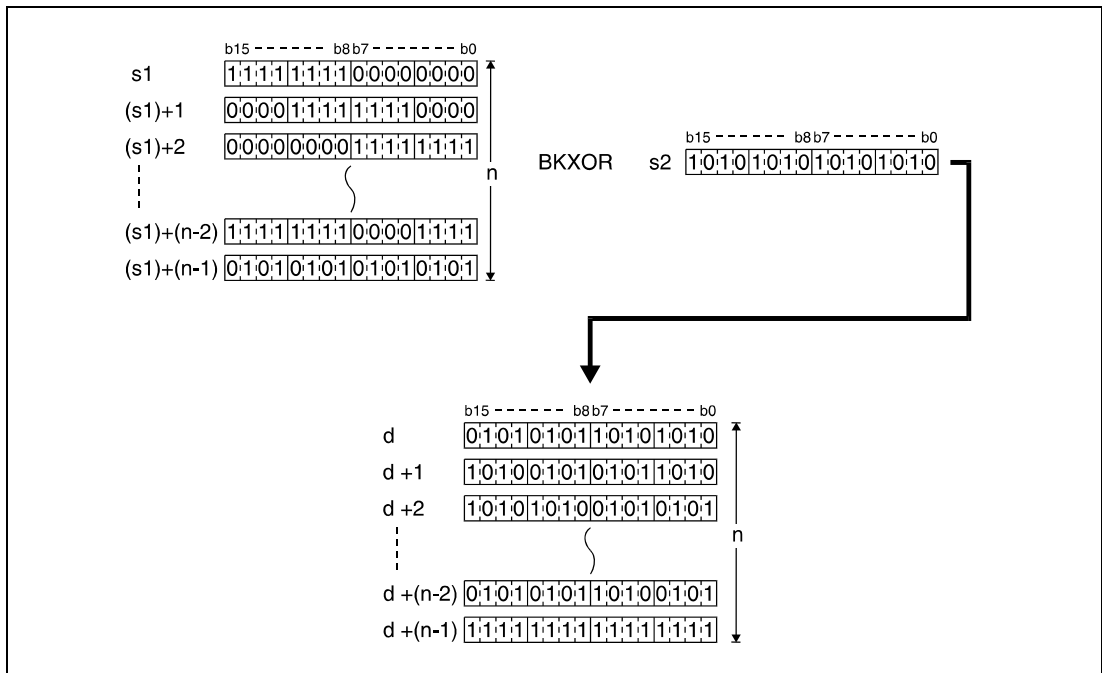
**Операция "исключающее ИЛИ" для 16-битных блоков**

**ВКXOR Поблочная операция "исключающее ИЛИ"**

Команда ВКXOR выполняет операцию "исключающее ИЛИ" для каждого из n 16-битных блоков, начиная с s1, и соответствующего блока из n 16-битных блоков, начиная с s2. Блок результата сохраняется, начиная с операнда, указанного в d. Количество блоков, в отношении которых выполняется операция, указано в n.



Указанная в s2 константа должна находиться в диапазоне между -32768 и 32767.



**Источники ошибок**

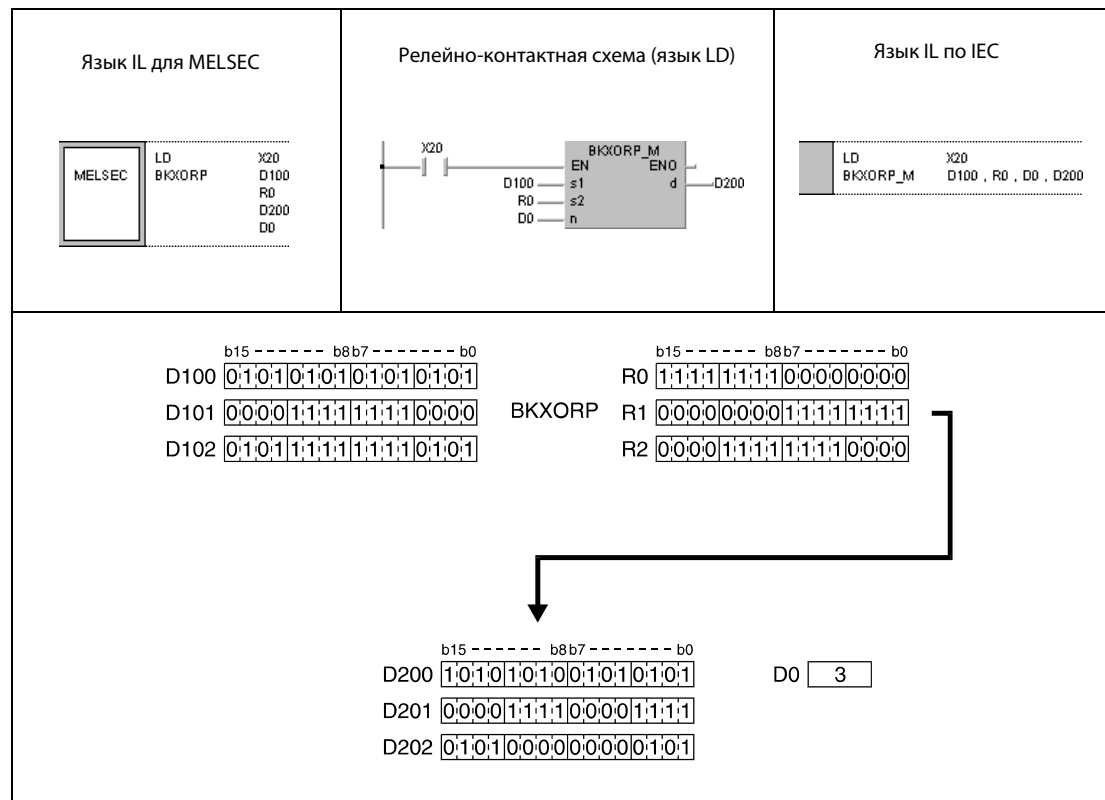
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в n количество блоков s1, s2 или d находится вне области операнда, предусмотренной для сохранения (код ошибки 4101).
- Предусмотренные для сохранения области s1, s2 или d перекрываются (код ошибки 4101).

**Пример**

**BKXORP**

Следующая программа при положительном фронте сигнала X20 выполняет операцию "исключающее ИЛИ" для данных в регистрах D100...D102 и данных в регистрах R0...R2. Результат сохраняется в регистрах с D200 по D202. Количество участвующих в операции 16-битных блоков данных (3) указано в D0.



7.1.7 WXNR, WXNRP, DXNR, DXNRP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

Операнды																Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки						
битовые				словные (16 бит)						Константы		Указатели		Уровень												
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1		Z						V	K	H (16#)	P	I	N
<b>WXNR</b>																K1 ↓ K4	5 ↓ 7	● ↓ ●		●						
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						●	●				
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●											
s1	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						●	●				
s2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						●	●				
d1		●	●	●	●	●	●	●	●	●	●	●	●	●	●											
<b>DXNR</b>																K1 ↓ K8	9 ↓ 1	● ↓ ●		●						
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						●	●				
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●											

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

Операнды MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□□G□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные							
<b>WXNR, WXNRP</b>											3
s	●	●	●	●	●	●	●	●	—		
d	●	●	●	●	●	●	●	—	—		
s1	●	●	●	●	●	●	●	●	—		
s2	●	●	●	●	●	●	●	●	—		
d1	●	●	●	●	●	●	●	—	—		
<b>DXNR, DXNRP</b>											4 <sup>1)</sup>
s	●	●	●	●	●	●	●	●	—		
d	●	●	●	●	●	●	●	—	—		
s1	●	●	●	●	●	●	●	●	—		
s2	●	●	●	●	●	●	●	●	—		
d1	●	●	●	●	●	●	●	—	—		

<sup>1</sup> Количество шагов программы зависит от типа центрального процессора и используемых операндов.

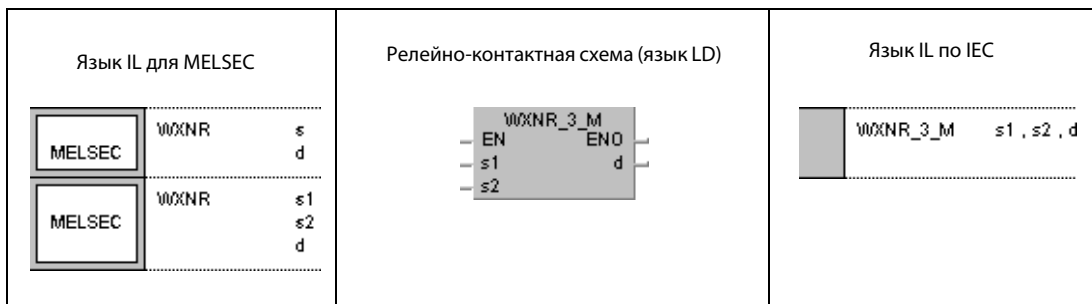
- При использовании процессоров QnA: 4
- В случае однопроцессорного модуля ЦП серии "Q": 3
- В случае многопроцессорного ЦПУ серии "Q", внутренних словных операндов (кроме регистра файлов ZR) или констант: 6
- В случае многопроцессорного ЦПУ серии "Q" и битовых операндов с адресом, кратным 16, имеющих обозначение битового блока K4 и не индексируемых: 6
- При использовании многопроцессорного ЦПУ серии "Q" и иных операндов кроме вышеуказанных: 4

<sup>2</sup> Количество шагов программы зависит от типа центрального процессора и используемых операндов.

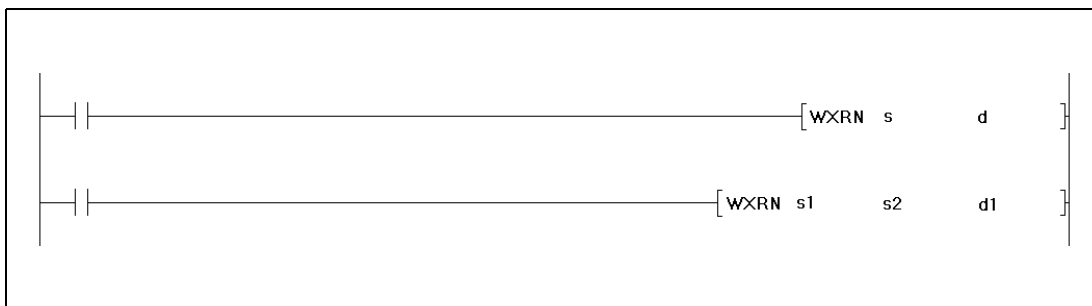
- При использовании процессоров QnA: 4
- В случае процессора "Q" и внутренних словных операндов (кроме регистра файлов ZR) или констант: 6
- В случае процессора "Q" и битовых операндов с адресом, кратным 16, имеющих обозначение битового блока K4 и не индексируемых: 6
- При использовании процессора Q и иных операндов кроме вышеуказанных: 4



**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Данные, в отношении которых выполняется операция "исключающее НЕ ИЛИ", или первый адрес операнда, в котором сохранены эти данные.	BIN, 16/32 бита
d		
s1	Данные, в отношении которых выполняется операция "исключающее НЕ ИЛИ", или первый адрес операнда, в котором сохранены эти данные.	
s2		
d (в случае WXNRP: d1)	Первый адрес операнда, в котором сохраняется результат.	

**Принцип действия**

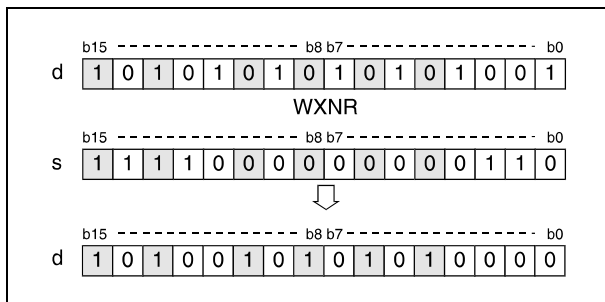
**Логическая операция "исключающее НЕ ИЛИ"**

**WXNR 16-битные данные**

Операция "исключающее НЕ ИЛИ" (по-английски: exclusive NOR) образует логическое произведение логической суммы двух входных переменных ( $Y = (\bar{A} + B) \times (A + \bar{B})$ ).

- 1-й вариант:

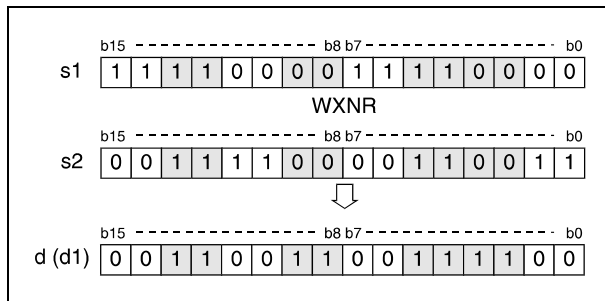
Логическая операция "исключающее НЕ ИЛИ" выполняется для 16-битных данных, указанных в s и d. Результат выводится в операнд, указанный в d.



● 2-й вариант:

Логическая операция "исключающее НЕ ИЛИ" выполняется для 16-битных данных, указанных в s1 и s2. Результат выводится в операнд, указанный в d.

При использовании команды WXNRP результат выводится в операнд, указанный в d1.

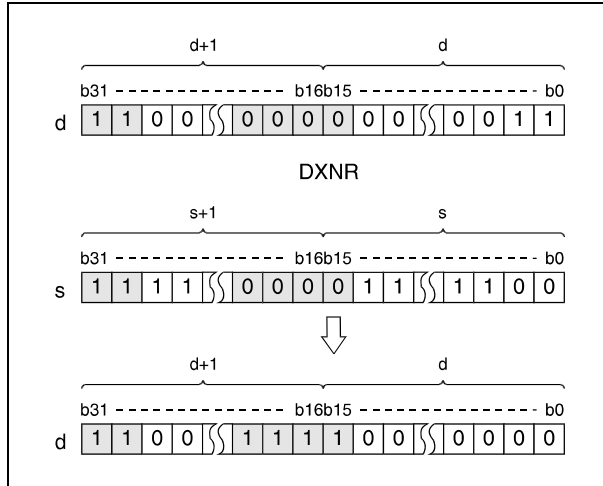


Биты сверх длины блока устанавливаются на 0. Например, если установлена длина блока K2, то старшие 8 битов (с b8 по b15) интерпретируются как биты с состоянием 0.

**DXNR 32-битные данные**

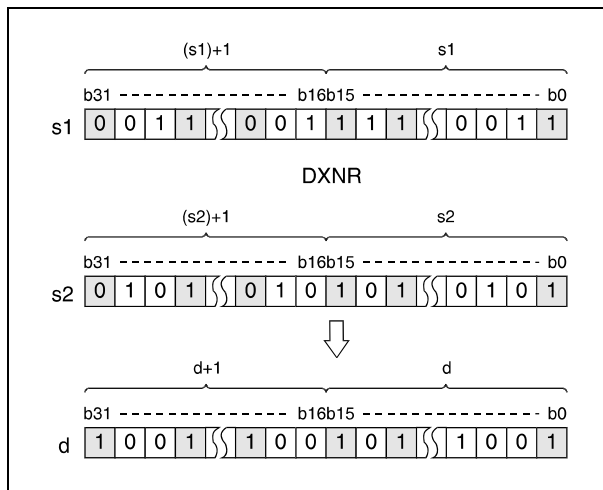
● 1-й вариант:

Логическая операция "исключающее НЕ ИЛИ" выполняется для 32-битных данных, указанных в s и d. Результат выводится в операнд, указанный в d.



● 2-й вариант (серия QnA/"System Q"):

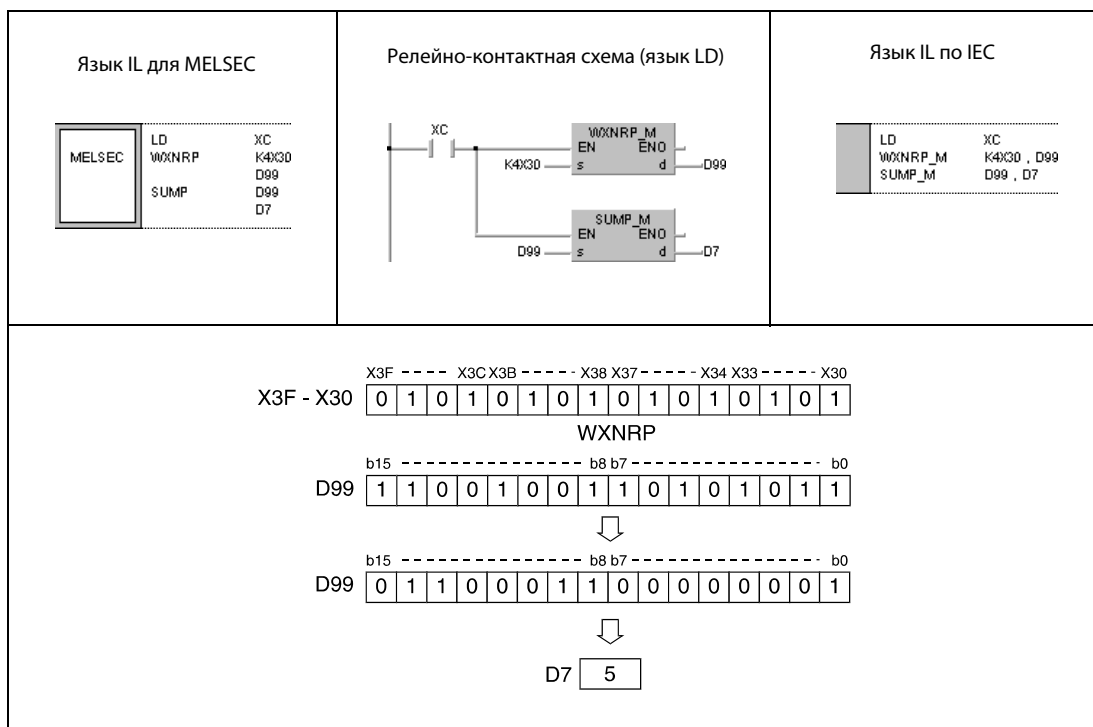
Логическая операция "исключающее НЕ ИЛИ" выполняется для 32-битных данных, указанных в s1 и s2. Результат выводится в операнд, указанный в d.



После выполнения соединения все биты, находящиеся вне области блока, устанавливаются на 0.

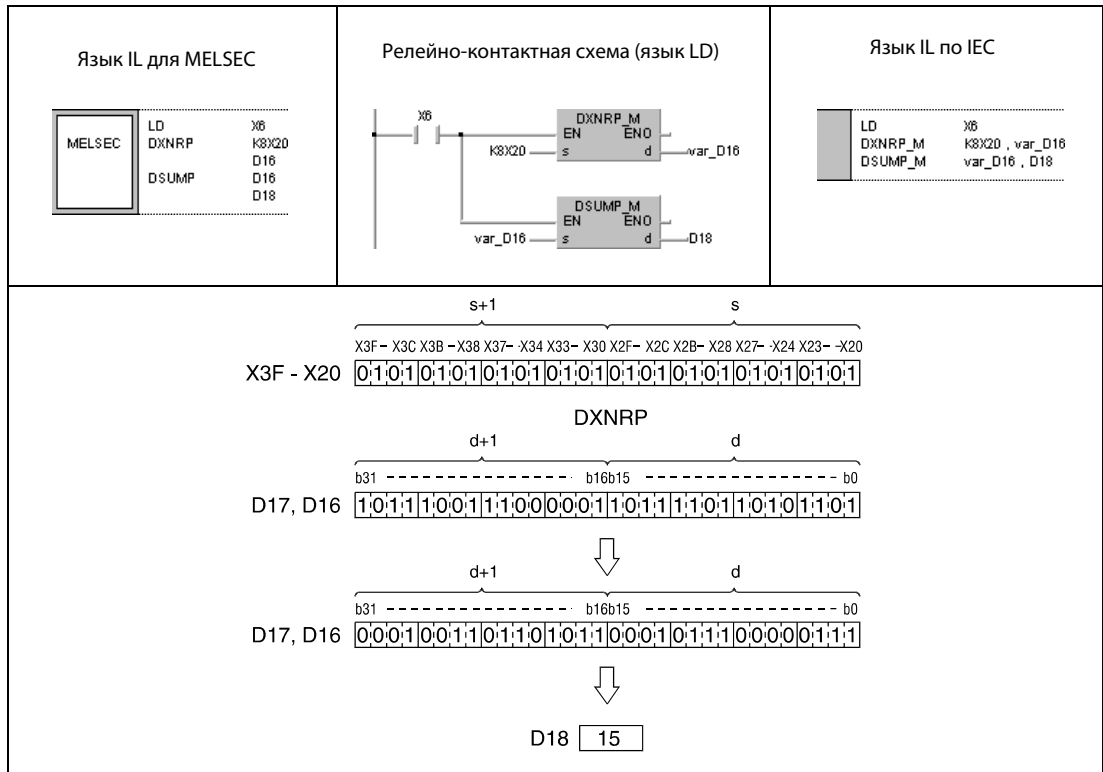
**Пример 1** WXNRP (s, d)

Следующая программа при положительном фронте сигнала XC сравнивает набор битов 16-битного значения входов X30...X3F со значением данных в D99 с помощью операции "исключающее НЕ ИЛИ" и результат операции снова сохраняет в D99. Количество установленных битов сохраняется в D7.



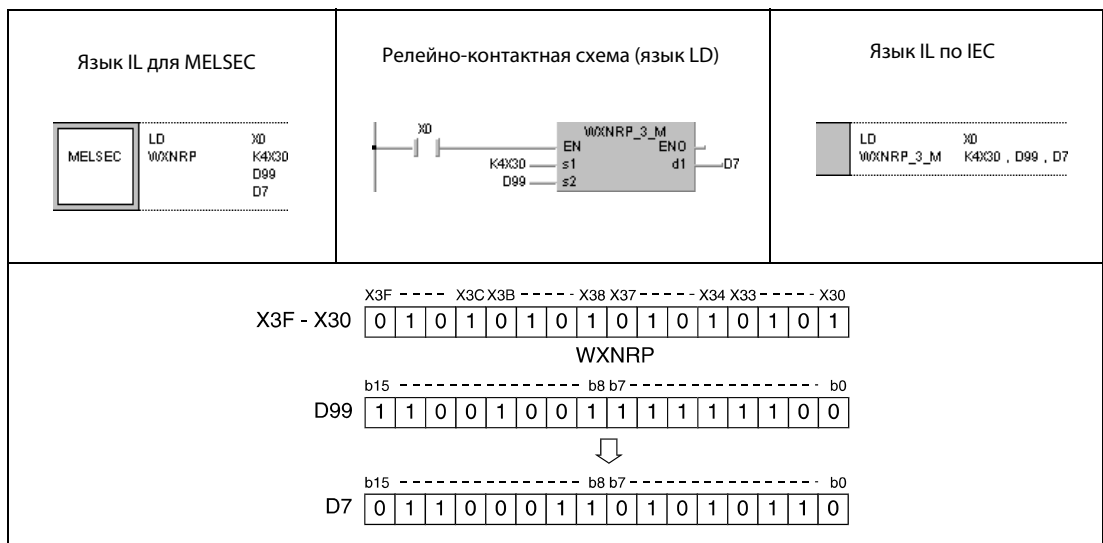
**Пример 2** DXNRP (s, d)

Следующая программа при положительном фронте сигнала X6 сравнивает набор битов 32-битного значения входов X20...X3F с данными в D16...D17 с помощью операции "исключающее НЕ ИЛИ" и результат операции снова сохраняет в D16 и D17. Количество установленных битов сохраняется в D18.



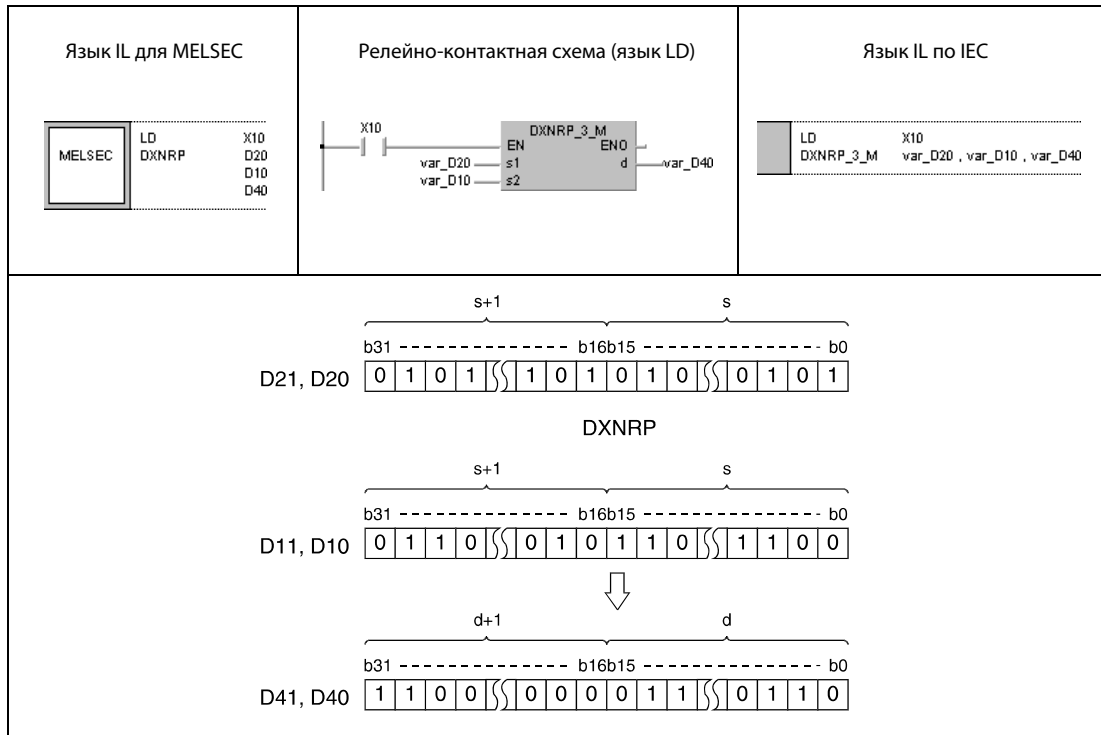
**Пример 3** WXNRP (s1, s2, d1)

В следующей программе при положительном фронте сигнала X0 выполняется операция "исключающее НЕ ИЛИ" для 16-битного значения входов X30...X3F и значения в D99. Результат сохраняется в D7.



**Пример 4** DXNRP (s1, s2, d)

В следующей программе при положительном фронте сигнала X10 выполняется операция "исключающее НЕ ИЛИ" для 32-битных данных в регистрах D20...D21 и данных в D10...D11. Результат сохраняется в D40 и D41.



**ПРИМЕЧАНИЕ** Без объявления переменной в заголовке программного компонента (POU) примеры программ 2 и 4 не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

**7.1.8 BKXNR, BKXNRP**

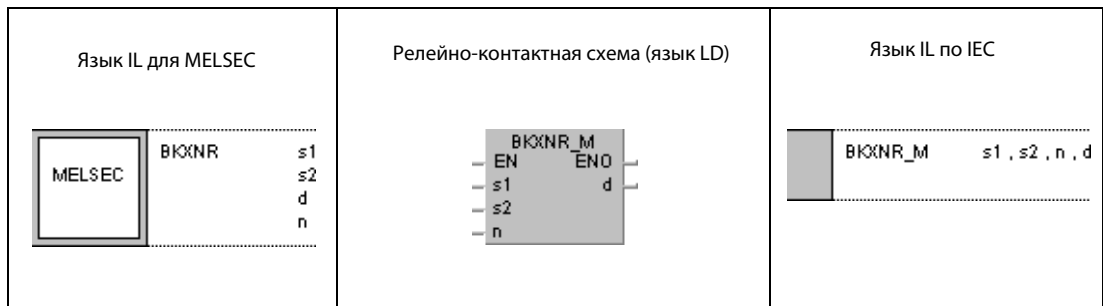
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	SM0	5	
s2	—	●	●	—	—	—	—	●			
d	—	●	●	—	—	—	—	—			
n	●	●	●	●	●	●	●	—			

GX IEC  
Developer



GX  
Developer



Переменные

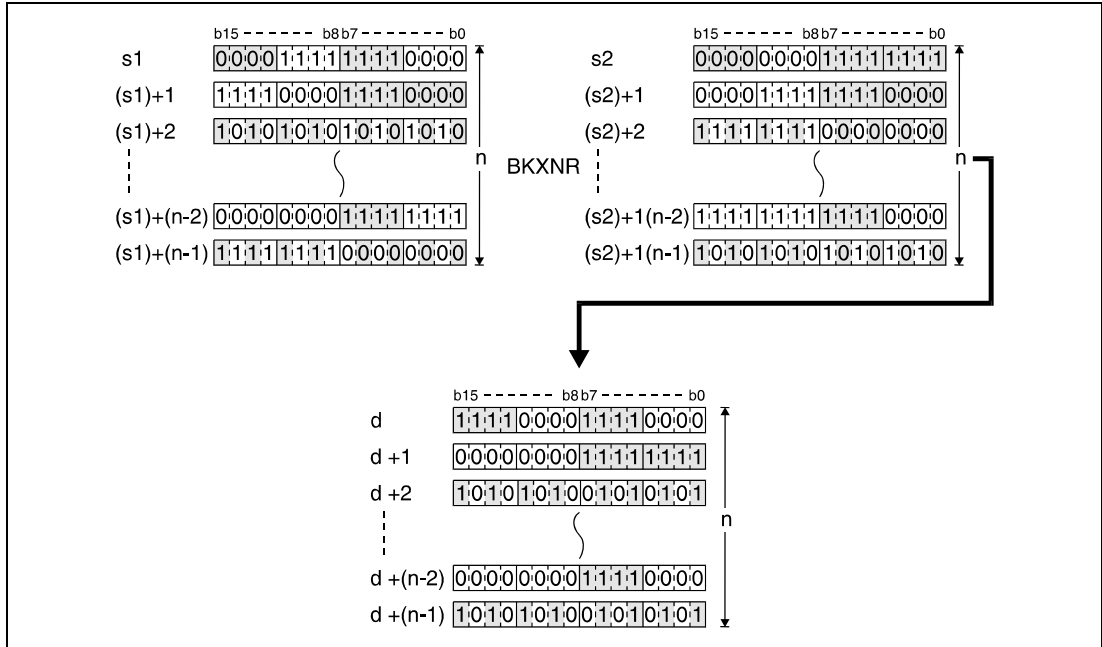
Операнд	Значение	Тип данных
s1	Первый адрес операнда, в котором сохранены данные для операции.	BIN, 16 бит
s2	Первый адрес данных, в отношении которых выполняется операция, или первый адрес операнда, в котором сохранены эти данные.	
d	Первый адрес операнда, в котором сохраняется результат операции.	
n	Количество блоков данных, в отношении которых выполняется операция "исключающее НЕ ИЛИ".	

**Принцип действия**

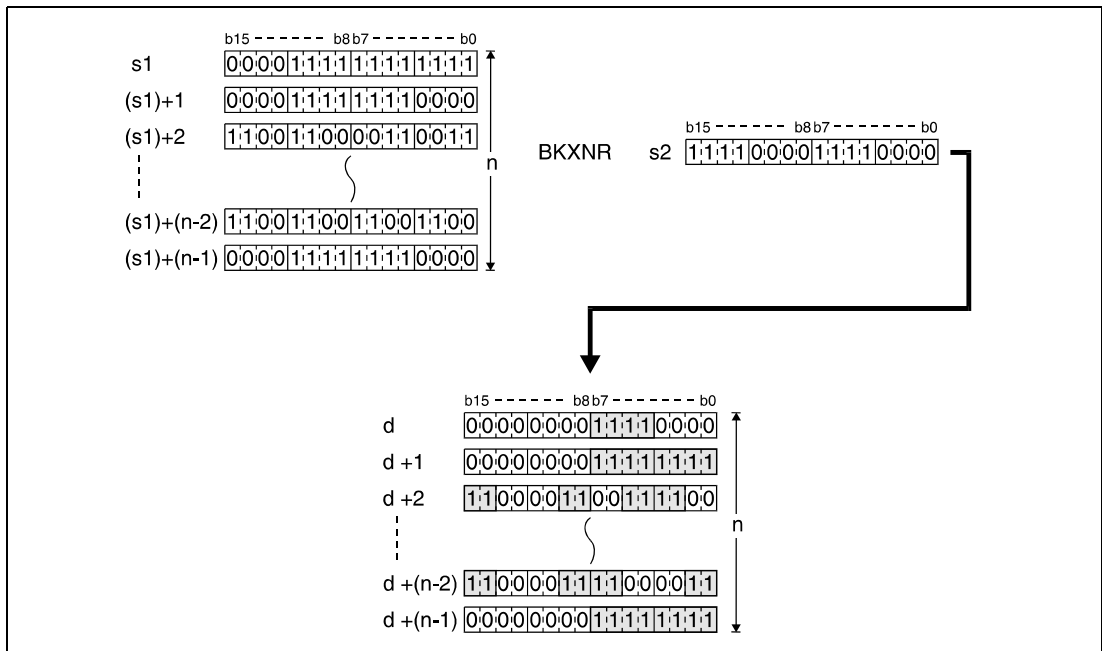
**Операции "исключающее НЕ ИЛИ" для 16-битных блоков данных**

**ВКXNR777 Поблочная операция "исключающее НЕ ИЛИ"**

Команда ВКXNR выполняет операцию "исключающее НЕ ИЛИ" в отношении каждого из n 16-битных блоков, начиная с s1, и соответствующего блока из n 16-битных блоков, начиная с s2. Соответствующий 16-битный блок результата сохраняется, начиная с операнда, указанного в d. Количество блоков, в отношении которых выполняется операция, указано в n.



Хранящаяся в s2 константа должна находиться в диапазоне между -32768 и 32767.





**Источники ошибок**

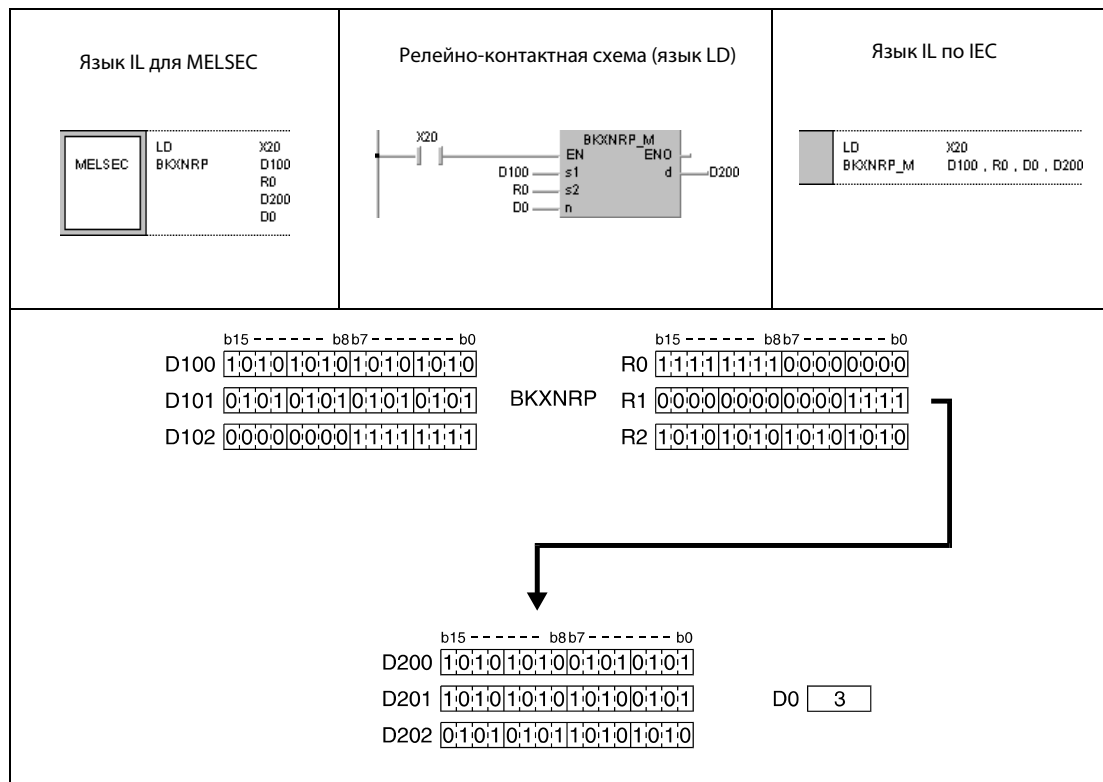
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в n количество блоков s1, s2 или d находится вне области операнда, предусмотренной для сохранения (код ошибки 4101).
- Предусмотренные для сохранения области s1, s2 или d перекрываются (код ошибки 4101).

**Пример**

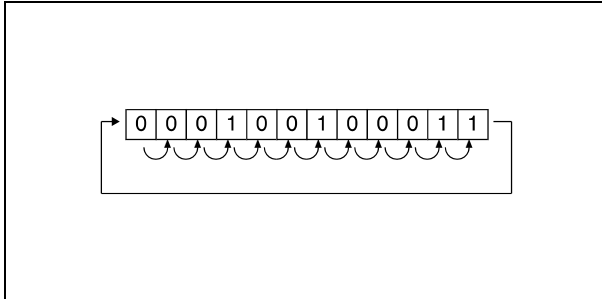
BKXNRP

Следующая программа при положительном фронте сигнала X20 выполняет операцию "исключающее НЕ ИЛИ" в отношении данных в регистрах D100...D102 и данных в регистрах R0...R2. Результат сохраняется в регистрах с D200 по D202. Количество участвующих в операции 16-битных блоков данных (3) указано в D0.



## 7.2 Команды вращения битов

Рассматриваемые далее команды служат для побитного вращения данных, промежуточно сохраненных в сумматорах и регистрах. Вращение может происходить как вправо, так и влево.



Команды вращения можно по выбору использовать с переносом или без. Команды применяются как для 16-битных, так и для 32-битных данных. В общей сложности имеются 16 различных команд вращения.

Функция	Команда MELSEC в редакторе MELSEC	Команда IEC в редакторе IEC
Вращение данных вправо (16 бит)	ROR	ROR_M
	RORP	RORP_M
	RORC	RORC_M
	RORCP	RORCP_M
Вращение данных влево (16 бит)	ROL	ROL_M
	ROLP	ROLP_M
	RCL	RCL_M
	RCLP	RCLP_M
Вращение данных вправо (32 бита)	DROR	DROR_M
	DRORP	DRORP_M
	DRCR	DRCR_M
	DRCRP	DRCRP_M
Вращение данных влево (32 бита)	DROL	DROL_M
	DROLP	DROLP_M
	DRCL	DRCL_M
	DRCLP	DRCLP_M

**ПРИМЕЧАНИЕ** В редакторах по стандарту IEC используйте стандартные команды IEC.

### 7.2.1 ROR, RORP, RCR, RCRP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

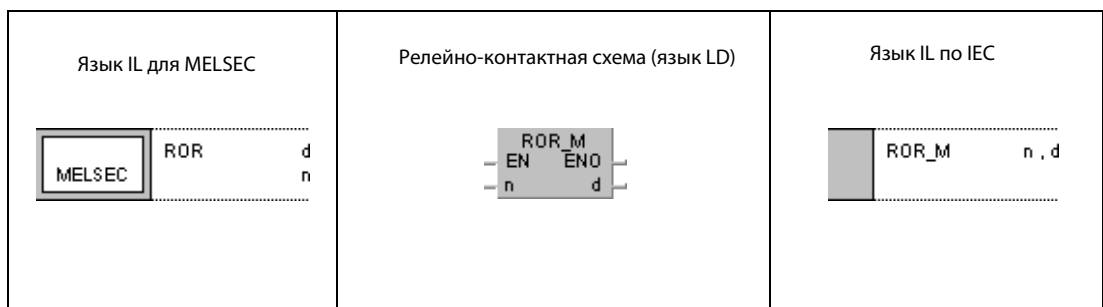
	Операнды															Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011					
	битовые					словные (16 бит)					Константы		Указатели		Уровень										
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V	K	H (16#)	P	I
n																●	●				3	1	●	●	●

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

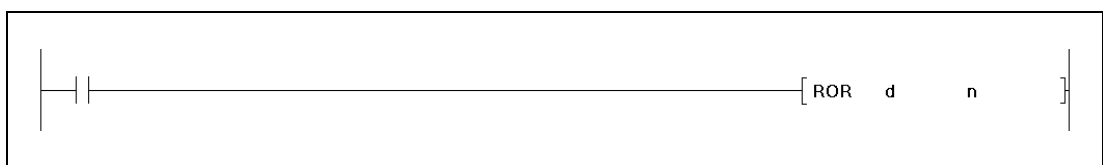
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
d	●	●	●	●	●	●	—	—	—	3	
n	●	●	●	●	●	●	●	—	—		

**GX IEC Developer**



**GX Developer**



**ПРИМЕЧАНИЕ**

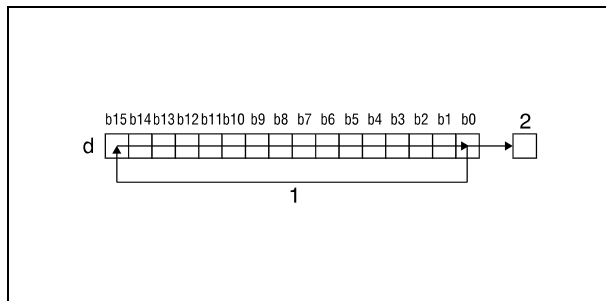
В случае серии "A" вращение всегда происходит в регистре A0. Поэтому при программировании этой команды в серии "A" отсутствует операнд d.

**Переменные**

Операнд	Значение	Тип данных
d	Начальный адрес операнда, в отношении которого выполняется операция вращения. Для серии "A" это всегда операнд A0.	BIN, 16 бит
n	Количество вращений (от 0 до 15)	

**Принцип действия****Вращение данных вправо (16 битов)****ROR Команда вращения без флага переноса**

Команда ROR вращает биты операнда, указанного в d (A0), вправо на n битов. Флаг переноса этой операцией не затрагивается. Флаг переноса (серия "A" = M9012, серии QnA/"System Q" = SM700) сохраняет значение последнего бита, перенесенного из разряда b0 в разряд b15 в результате вращения.

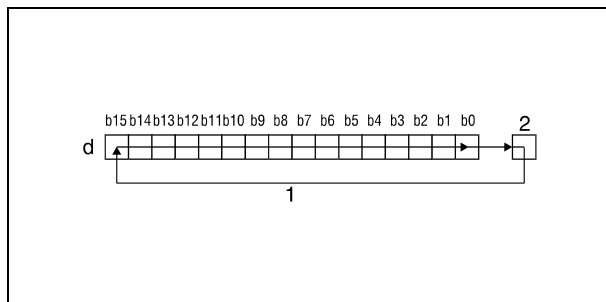


<sup>1</sup> вращение на n битов

<sup>2</sup> флаг переноса

**RCR Команда вращения с флагом переноса**

Команда RCR вращает биты операнда, указанного в d (A0), вправо на n битов. В это вращение вовлекается и флаг переноса. При этом флаг переноса (серия "A" = M9012, серии "Q"/"System Q" = SM700) принимает значение бита, смещенного на n разрядов. Имевшееся перед вращением содержимое флага переноса (0 или 1) перемещается в d (A0) на n разрядов вправо, начиная от разряда b15.



<sup>1</sup> вращение на n битов

<sup>2</sup> флаг переноса

**ПРИМЕЧАНИЯ** Только серии "Q" и "System Q":

Если в d указан битовый операнд, то операция вращения выполняется с операндом указанной ширины. Количество разрядов, на которые вращаются биты, определяется как остаток от следующего деления:

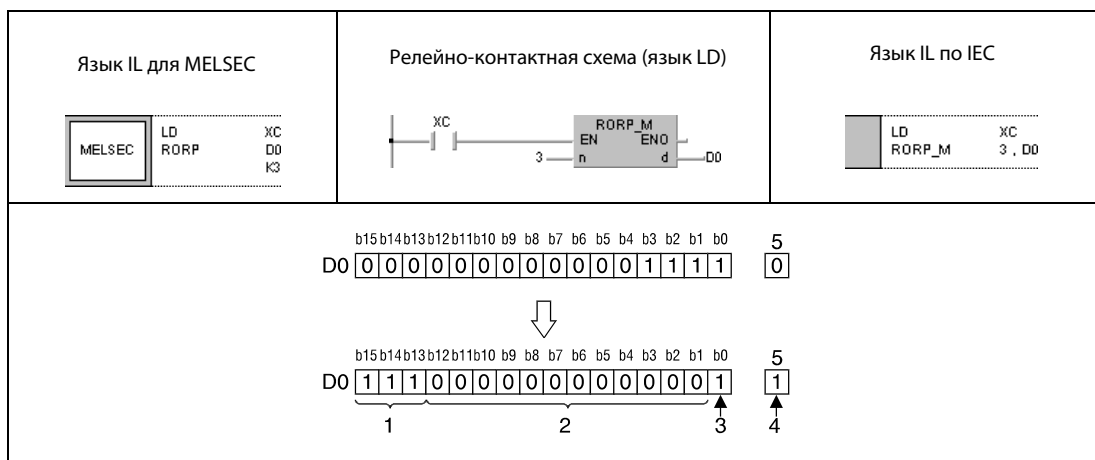
*Количество вращений n/количество битов*

Например, 16-кратное вращение 12 битов соответствовало бы вращению на 4 бита, так как остаток от деления 16/12 равен 4. Это происходит потому, что после 12-кратного вращения бит x из 12 битов снова оказывается на том самом месте, на котором он находился перед вращением.

Поэтому указывайте для n значение от 0 до 15.

**Пример 1** RORP (серии "Q" и "System Q")

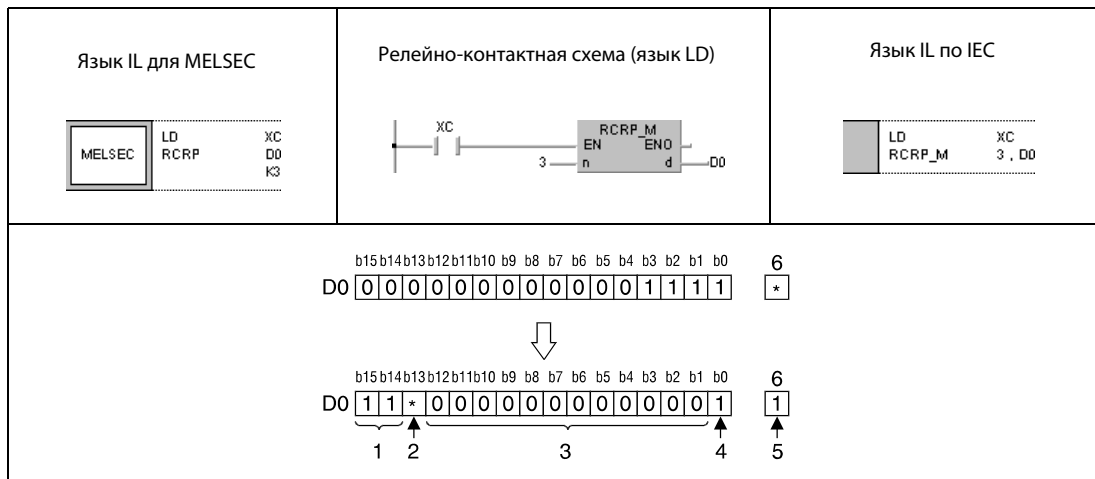
Следующая программа при положительном фронте сигнала XС вращает содержимое D0 на 3 бита вправо.



- 1 состояние битов b0–b2 перед вращением
- 2 состояние битов b4–b15 перед вращением
- 3 состояние бита b3 перед вращением
- 4 состояние бита b2 перед вращением
- 5 флаг переноса

**Пример 2** RCRP (серии "Q" и "System Q")

Следующая программа при положительном фронте сигнала XС вращает содержимое D0, в том числе флаг переноса SM700, на 3 бита вправо. Состояние SM700 перед вращением (0/1) сдвигается на 3 разряда вправо.



- 1 состояние битов b1 и b0 перед вращением
- 2 состояние флага переноса перед вращением
- 3 состояние битов b4–b15 перед вращением
- 4 состояние бита b3 перед вращением
- 5 состояние бита b2 перед вращением
- 6 флаг переноса

7.2.2 ROL, ROLP, RCL, RCLP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

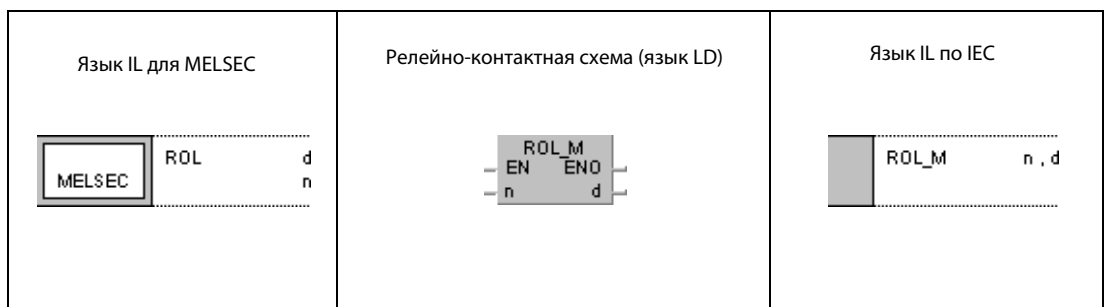
Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011						
битовые						словные (16 бит)						Константы		Указа-тели	Уро-вень											
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						K	H (16#)	P	I	N	
n																	●	●				3	1	●	●	●

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

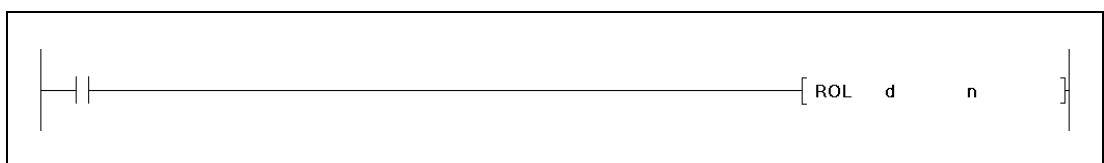
Операнды MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные							
d	●	●	●	●	●	●	—	—	—	—	3
n	●	●	●	●	●	●	●	—	—	—	

GX IEC Developer



GX Developer



**ПРИМЕЧАНИЕ**

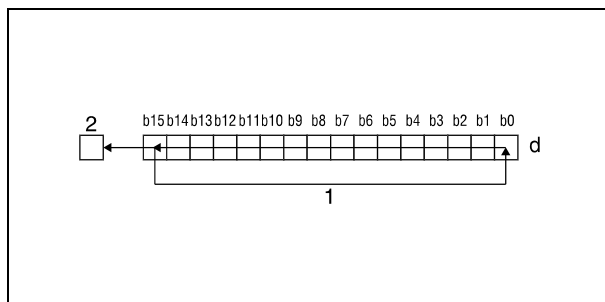
В случае серии "А" вращение всегда происходит в регистре A0. Поэтому при программировании этой команды в серии "А" отсутствует операнд d.

Переменные

Операнд	Значение	Тип данных
d	Начальный адрес операнда, в отношении которого выполняется операция вращения. Для серии "А" это всегда операнд A0.	BIN, 16 бит
n	Количество вращений (от 0 до 15)	

**Принцип действия****Вращение данных влево (16 бит)****ROL Команда вращения без флага переноса**

Команда ROL вращает биты данных в d (A0) на n битов влево. Флаг переноса этой операцией не затрагивается. Флаг переноса (серия "A" = M9012, серии "Q"/"System Q" = SM700) принимает значение последнего бита, перенесенного из разряда b15 в разряд b0 в ходе операции вращения.

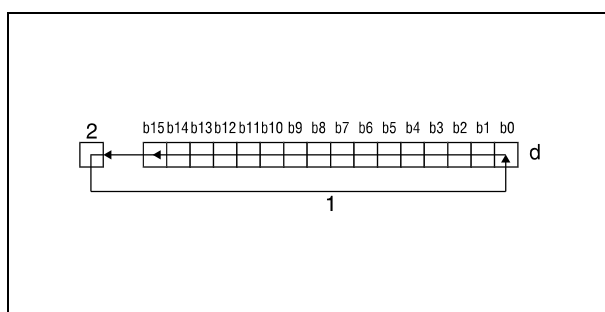


<sup>1</sup> вращение на n битов

<sup>2</sup> флаг переноса

**RCL Команда вращения с флагом переноса**

Команда RCL вращает биты данных в d (A0) на n битов влево. В это вращение вовлекается и флаг переноса. При этом флаг переноса (серия "A" = M9012, серии "Q"/"System Q" = SM700) принимает значение бита, смещенного на n разрядов. Имевшееся перед вращением содержимое флага переноса (0 или 1) перемещается дальше в d (A0) на n разрядов влево, начиная с разряда b0.



<sup>1</sup> вращение на n битов

<sup>2</sup> флаг переноса

**ПРИМЕЧАНИЯ** Только серии "Q" и "System Q":

Если в d указан битовый операнд, то операция вращения выполняется с операндом указанной ширины. Количество разрядов, на которые вращаются биты, определяется как остаток от следующего деления:

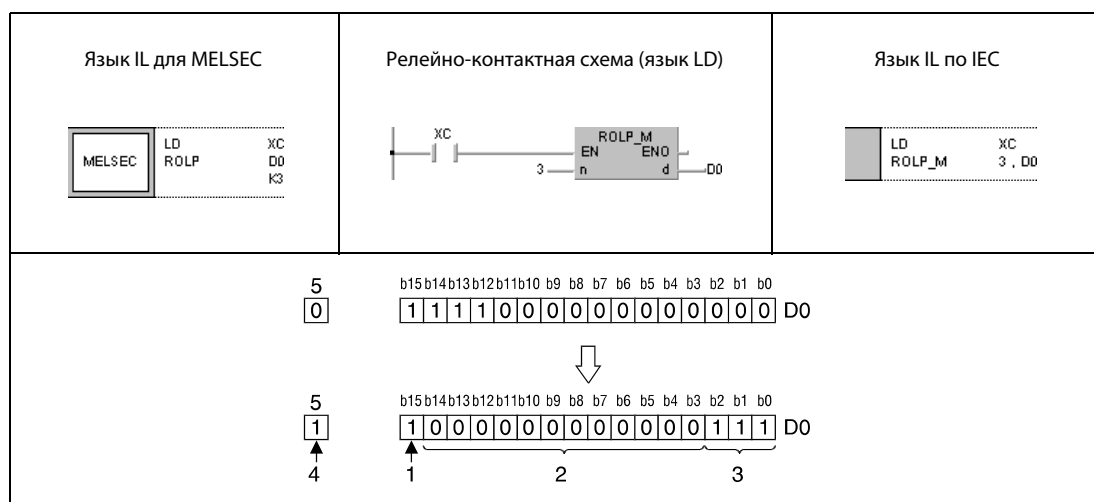
*Количество вращений n/количество битов*

Например, 16-кратное вращение 12 битов соответствовало бы вращению на 4 бита, так как остаток от деления 16/12 равен 4. Это происходит потому, что после 12-кратного вращения бит x из 12 битов снова оказывается на том самом месте, на котором он находился перед вращением.

Поэтому указывайте для n значение от 0 до 15.

**Пример 1** ROLP (серии "Q" и "System Q")

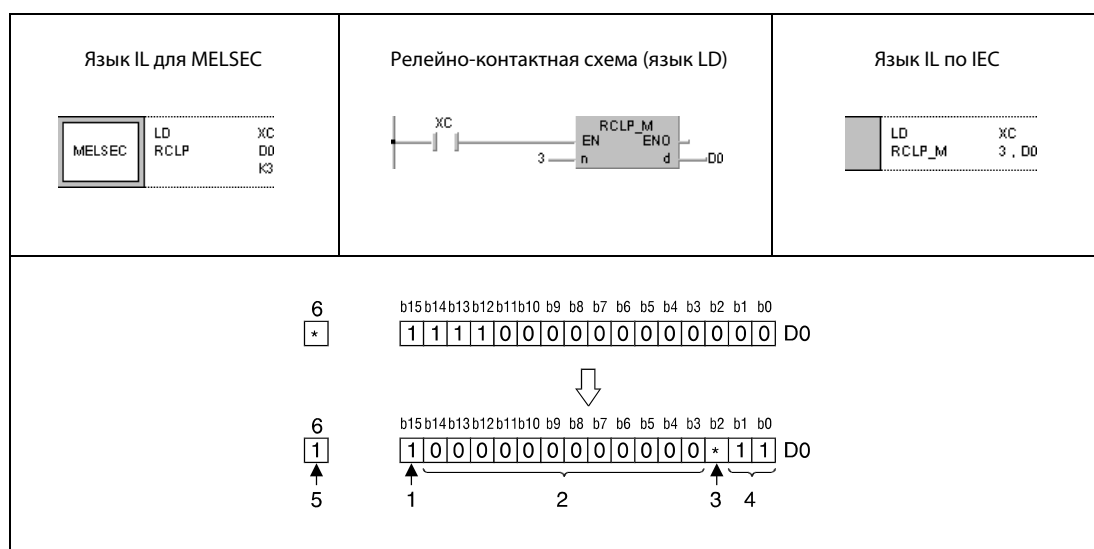
Следующая программа при положительном фронте сигнала XС вращает содержимое D0 на 3 бита влево.



- 1 состояние бита b12 перед вращением
- 2 состояние битов b11–b0 перед вращением
- 3 состояние битов b15–b13 перед вращением
- 4 состояние бита b12 перед вращением
- 5 флаг переноса

**Пример 2** RCLP (серии "Q" и "System Q")

Следующая программа при положительном фронте сигнала XС вращает содержимое D0 и флага переноса SM700 на 3 бита влево. Состояние SM700 перед вращением (0/1) сдвигается влево на 3 разряда.



- 1 состояние бита b12 перед вращением
- 2 состояние битов b11–b0 перед вращением
- 3 состояние флага переноса
- 4 состояние битов b14 и b15 перед вращением
- 5 состояние флага переноса перед вращением
- 6 флаг переноса



### 7.2.3 DROR, DRORP, DRCR, DRCRP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

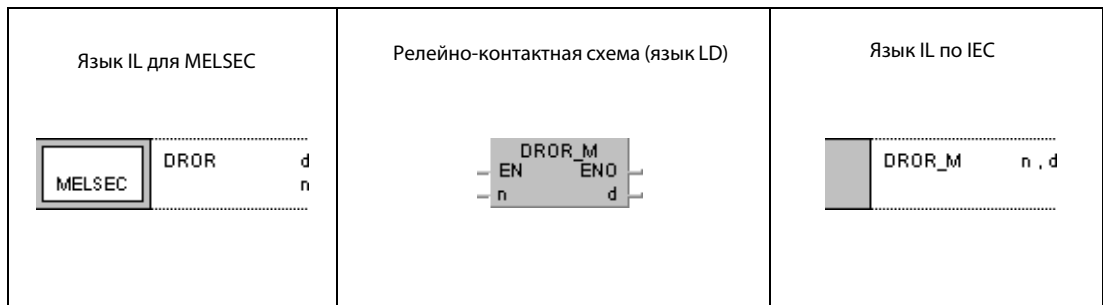
	Операнды															Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011							
	битовые					словные (16 бит)					Константы		Указатели		Уровень												
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V	K	H (16#)	P	I	N	
n																	●	●				3	1	●	●	●	●

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

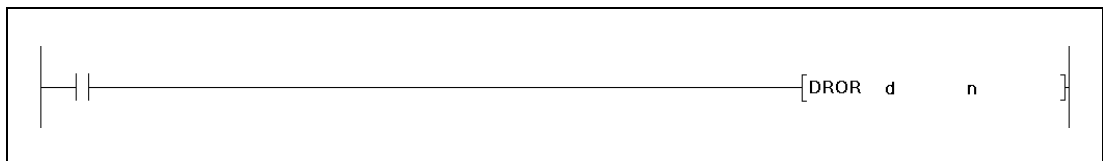
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
d	●	●	●	●	●	●	●	—	—	—	3
n	●	●	●	●	●	●	●	●	—	—	

**GX IEC Developer**



**GX Developer**



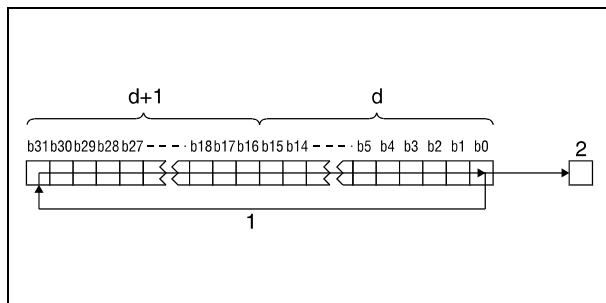
**ПРИМЕЧАНИЕ** В случае серии "A" вращение всегда происходит в регистрах A0 и A1. Поэтому при программировании этой команды в серии "A" отсутствует операнд d.

**Переменные**

Операнд	Значение	Тип данных
d	Первый адрес операнда, в отношении которого выполняется операция вращения. Для серии "A" это всегда операнды A0 и A1.	BIN, 32 бита
n	Количество вращений (от 0 до 31)	BIN, 16 бит

**Принцип действия****Вращение данных вправо (32 бита)****DROR Команда вращения без флага переноса**

Команда DROR вращает биты данных в  $d$  (A0, A1) на  $n$  битов вправо. Флаг переноса этой операцией не затрагивается. Флаг переноса (серия "A" = M9012, серии "Q"/"System Q" = SM700) принимает значение последнего бита, перенесенного из  $b0$  в  $b31$  в ходе операции вращения.

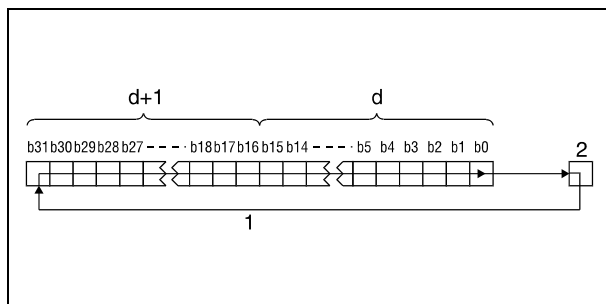


<sup>1</sup> вращение на  $n$  битов

<sup>2</sup> флаг переноса

**DRCR Команда вращения с флагом переноса**

Команда DRCR вращает биты данных в  $d$  (A0, A1) на  $n$  битов вправо. В это вращение вовлекается и флаг переноса. При этом флаг переноса (серия "A" = M9012, серии "Q"/"System Q" = SM700) принимает значение бита, смещенного на  $n$  разрядов. Имевшееся перед вращением содержимое флага переноса (0 или 1) перемещается дальше в  $d$  (A0, A1) на  $n$  разрядов вправо, начиная с разряда  $b31$ .



<sup>1</sup> вращение на  $n$  битов

<sup>2</sup> флаг переноса

**ПРИМЕЧАНИЯ** Только серии "Q" и "System Q"

Если в  $d$  указан битовый операнд, то операция вращения выполняется с операндом указанной ширины. Количество разрядов, на которые вращаются биты, определяется как остаток от следующего деления:

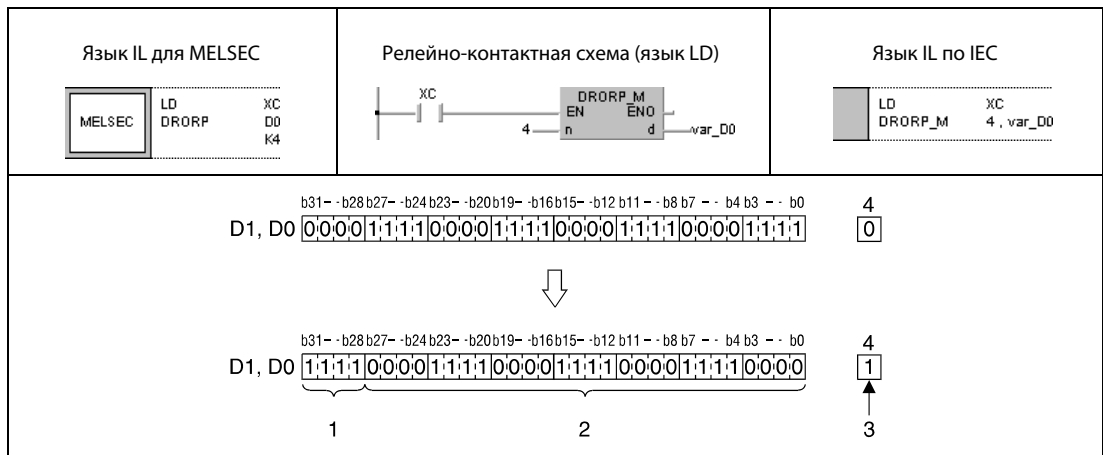
$\text{Количество вращений } n / \text{количество битов}$

Например, 31 вращение 24 битов соответствовало бы вращению на 7 битов, так как остаток от деления  $31/24$  равен 7. Это происходит потому, что после 24 вращений бит  $x$  из 24 битов снова оказывается на том самом месте, на котором он находился перед вращением.

Указывайте для  $n$  значение от 0 до 31.

**Пример 1** DRORP (серии "Q" и "System Q")

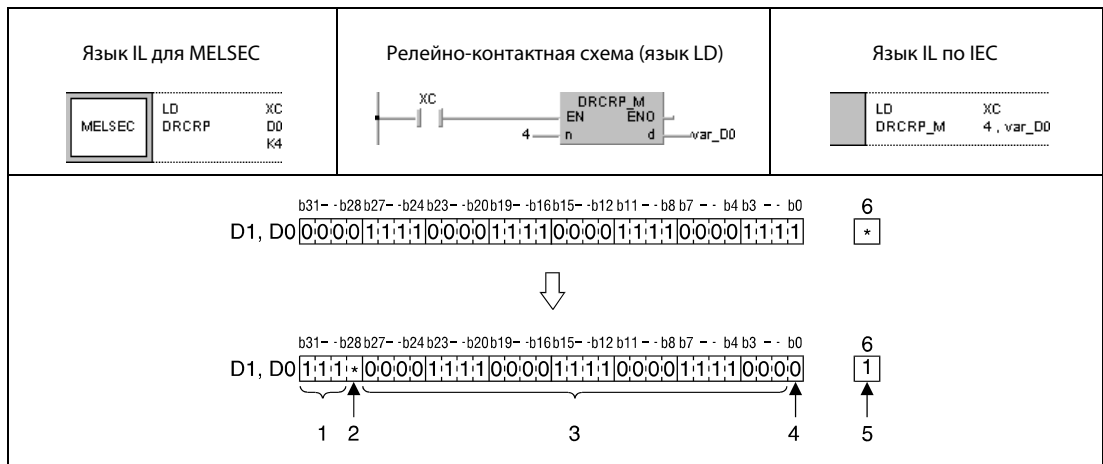
Следующая программа при положительном фронте сигнала XC вращает содержимое D0 и D1 на 4 бита вправо.



- 1 состояние битов b3–b0 перед вращением
- 2 состояние битов b31–b4 перед вращением
- 3 состояние бита b3 перед вращением
- 4 флаг переноса

**Пример 2** DRCRP (серии "Q" и "System Q")

Следующая программа при положительном фронте сигнала XC вращает содержимое D0 и D1 и флага переноса SM700 на 4 бита вправо. Состояние SM700 перед вращением (0/1) сдвигается вправо на 4 разряда.



- 1 состояние битов b2–b0 перед вращением
- 2 состояние флага переноса перед вращением
- 3 состояние битов b5–b31 перед вращением
- 4 состояние бита b4 перед вращением
- 5 состояние бита b3 перед вращением
- 6 флаг переноса

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

7.2.4 DROL, DROLP, DRCL, DRCLP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	

Операнды MELSEC A

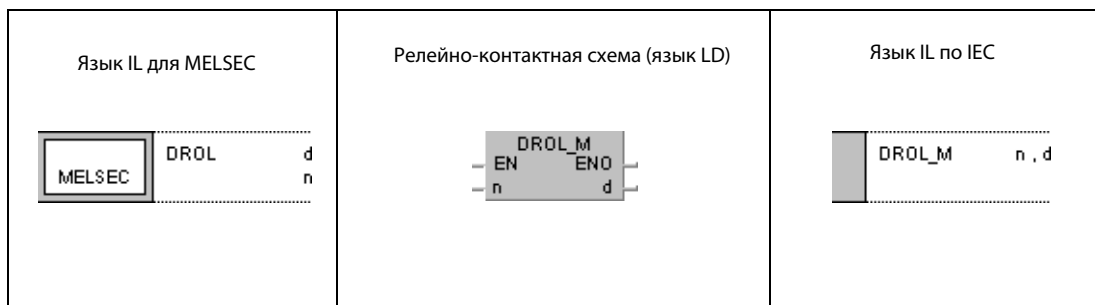
	Операнды															Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011					
	битовые					словные (16 бит)					Константы		Указатели		Уровень										
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V	K	H (16#)	P	I
n																●	●				3	1	●	●	●

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

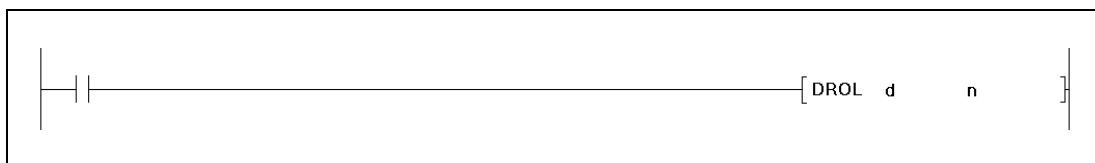
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□G□	Индексные регистры Zn	Константы E			Иные
	битовые	словные		битовые	словные						
d	●	●	●	●	●	●	●	—	—	—	3
n	●	●	●	●	●	●	●	●	—	—	

GX IEC Developer



GX Developer



ПРИМЕЧАНИЕ

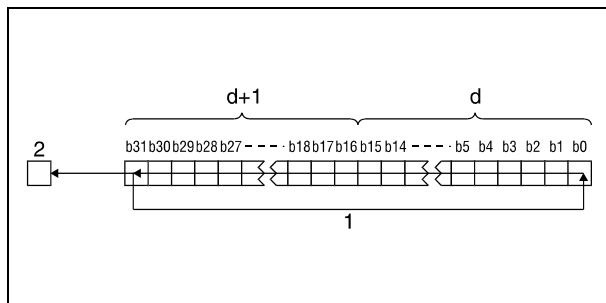
В случае серии "A" вращение всегда происходит в регистрах A0 и A1. Поэтому при программировании этой команды в серии "A" отсутствует операнд d.

Переменные

Операнд	Значение	Тип данных
d	Первый адрес операнда, в отношении которого выполняется операция вращения. Для серии "A" это всегда операнды A0 и A1.	BIN, 32 бита
n	Количество вращений (от 0 до 31)	BIN, 16 бит

**Принцип действия****Вращение данных влево (32 бита)****DROL Команда вращения без флага переноса**

Команда DROL вращает биты данных в *d* (A0, A1) на *n* битов влево. Флаг переноса этой операцией не затрагивается. Флаг переноса (серия "A" = M9012, серии "Q"/"System Q" = SM700) принимает значение последнего бита, перенесенного из *b*31 в *b*0 в ходе операции вращения.

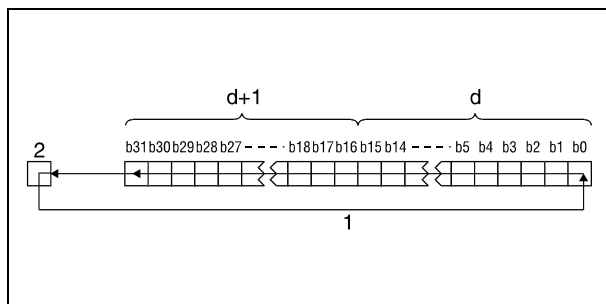


<sup>1</sup> вращение на *n* битов

<sup>2</sup> флаг переноса

**DRCL Команда вращения с флагом переноса**

Команда DRCL вращает биты данных в *d* (A0, A1) на *n* битов влево. В это вращение вовлекается и флаг переноса. При этом флаг переноса (серия "A" = M9012, серии "Q"/"System Q" = SM700) принимает значение бита, смещенного на *n* разрядов. Имевшееся перед вращением содержимое флага переноса (0 или 1) перемещается дальше в *d* (A0, A1) на *n* разрядов влево, начиная с разряда *b*31.



<sup>1</sup> вращение на *n* битов

<sup>2</sup> флаг переноса

**ПРИМЕЧАНИЯ** Только серия Q и System Q

Если в *d* указан битовый операнд, то операция вращения выполняется с операндом указанной ширины. Количество разрядов, на которые вращаются биты, определяется как остаток от следующего деления:

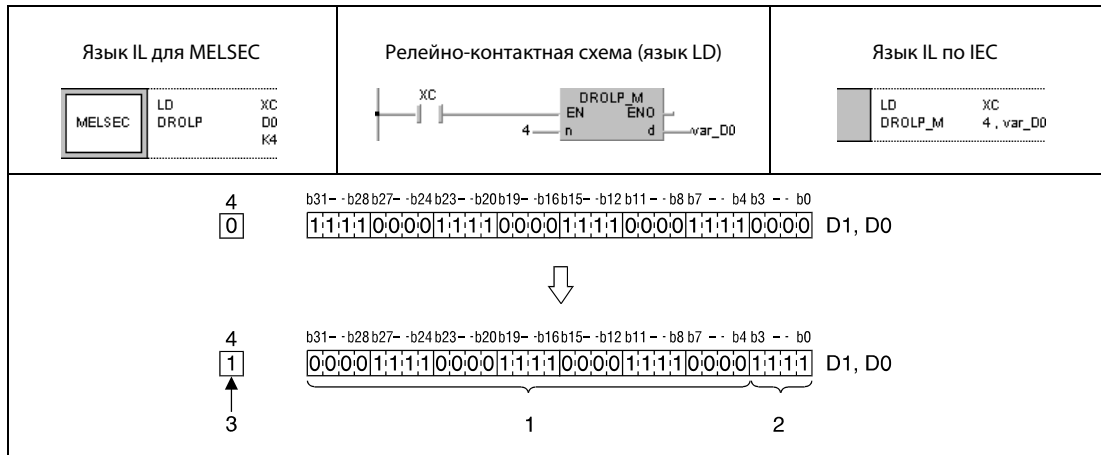
*Количество вращений n* / количество битов

Например, 31 вращение 24 битов соответствовало бы вращению на 7 битов, так как остаток от деления 31/24 равен 7. Это происходит потому, что после 24 вращений бит *x* из 24 битов снова оказывается на том самом месте, на котором он находился перед вращением.

Указывайте для *n* значение от 0 до 31.

**Пример 1** DROLP (серии "Q" и "System Q")

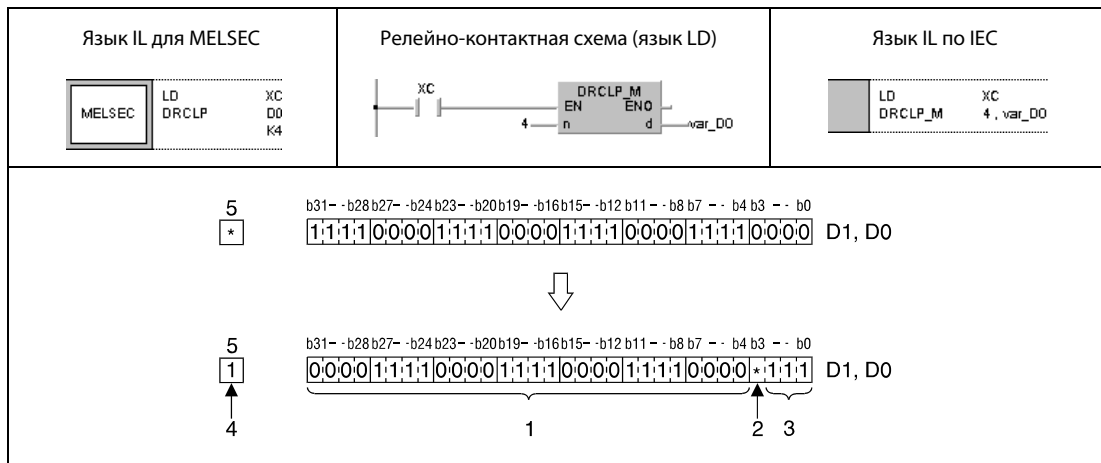
Следующая программа при положительном фронте сигнала XC вращает содержимое D0 и D1 на 4 бита влево.



- 1 состояние битов b27–b0 перед вращением
- 2 состояние битов b31–b28 перед вращением
- 3 состояние бита b28 перед вращением
- 4 флаг переноса

**Пример 2** DRCLP (серии "Q" и "System Q")

Следующая программа при положительном фронте сигнала XC вращает содержимое D0 и D1 и флага переноса SM700 на 4 бита влево. Состояние SM700 перед вращением (0/1) сдвигается на 4 разряда влево.



- 1 состояние битов b27–b0 перед вращением
- 2 состояние флага переноса перед вращением
- 3 состояние битов b31–b29 перед вращением
- 4 состояние бита b28 перед вращением
- 5 флаг переноса

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

## 7.3 Команды сдвига

Команды сдвига (SHIFT) выполняют побитный или поблочный сдвиг данных в пределах слова данных. Сдвиг может выполняться как вправо, так и влево.

В общей сложности имеются 12 команд сдвига.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе IEC
Сдвиг 16-битного слова данных на n битов	SFR	SFR_M
	SFRP	SFRP_M
	SFL	SFL_M
	SFLP	SFLP_M
Сдвиг n битовых операндов на 1 бит	BSFR	BSFR_M
	BSFRP	BSFRP_M
	BSFL	BSFL_M
	BSFLP	BSFLP_M
Сдвиг n словных операндов на 1 разряд	DSFR	DSFR_M
	DSFRP	DSFRP_M
	DSFL	DSFL_M
	DSFLP	DSFLP_M

**ПРИМЕЧАНИЕ** В редакторах по стандарту IEC используйте стандартные команды IEC.

7.3.1 SFR, SFRP, SFL, SFLP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

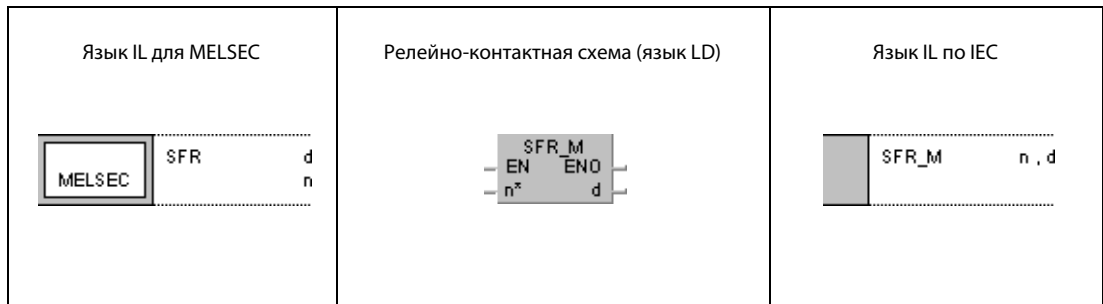
	Операнды															Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки						
	битовые					словные (16 бит)					Константы		Указатели		Уровень											
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1							Z	V	K	H (16#)	P	I
d	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●						K1 ↓ K4	3	●	●	●	●
n																●	●									

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

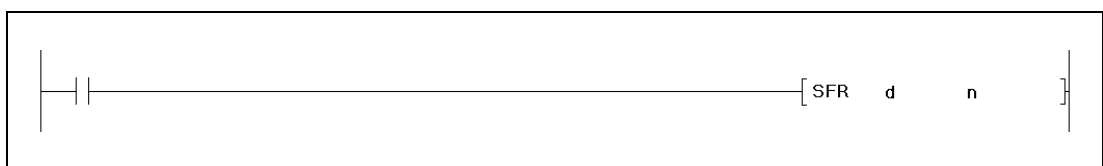
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
d	●	●	●	●	●	●	●	—	—	—	3
n	●	●	●	●	●	●	●	—	—		

GX IEC Developer



GX Developer



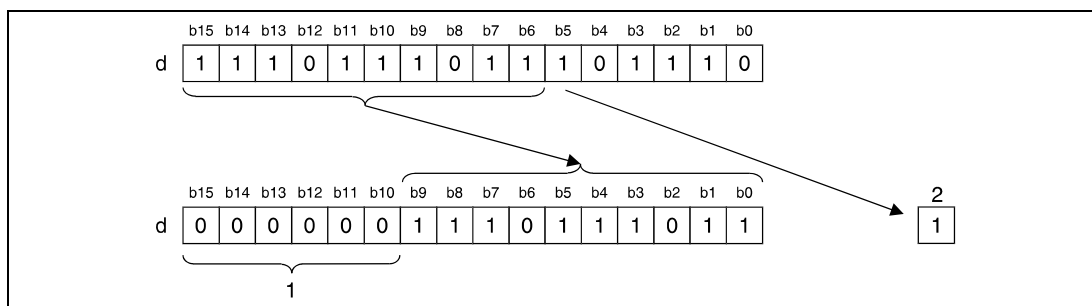
Переменные

Операнд	Значение	Тип данных
d	Первый адрес операнда, в котором сохранены сдвигаемые данные.	BIN, 16 бит
n	Количество сдвигов (от 0 до 15)	



**Принцип действия****Сдвиг 16-битного слова данных на n битов****SFR Сдвиг вправо**

Команда SFR побитно сдвигает указанное в d 16-битное слово данных на n битов вправо.



<sup>1</sup> в эти биты записываются нули

<sup>2</sup> флаг переноса

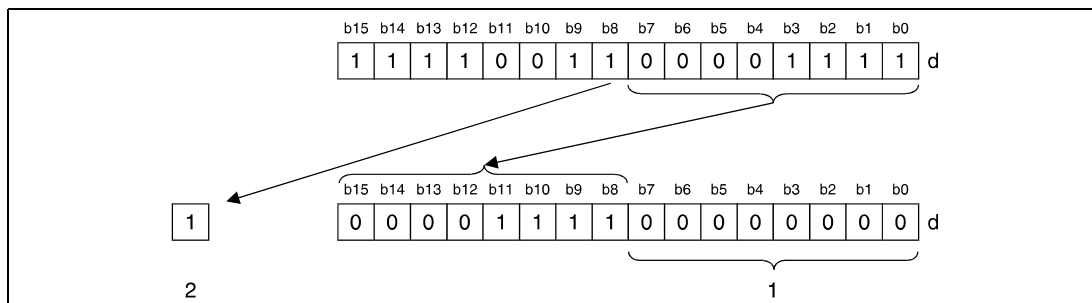
Старшие n битов, начиная с бита b15, устанавливаются на 0. n-ый сдвигаемый бит (b (n – 1)) записывается во флаг переноса (серия "А" = M9012, серии "Q"/"System Q" = SM700).

В случае таймеров или счетчиков сдвигается фактическое значение (состояние счета). Сдвиг заданного значения (уставки) не возможен.

При использовании битовых операндов возможен сдвиг в пределах операнда на указанное число битов (см. пример 1).

**SFL Сдвиг влево**

Команда SFL побитно сдвигает указанное в d 16-битное слово данных на n битов влево.



<sup>1</sup> в эти биты записываются нули.

<sup>2</sup> флаг переноса

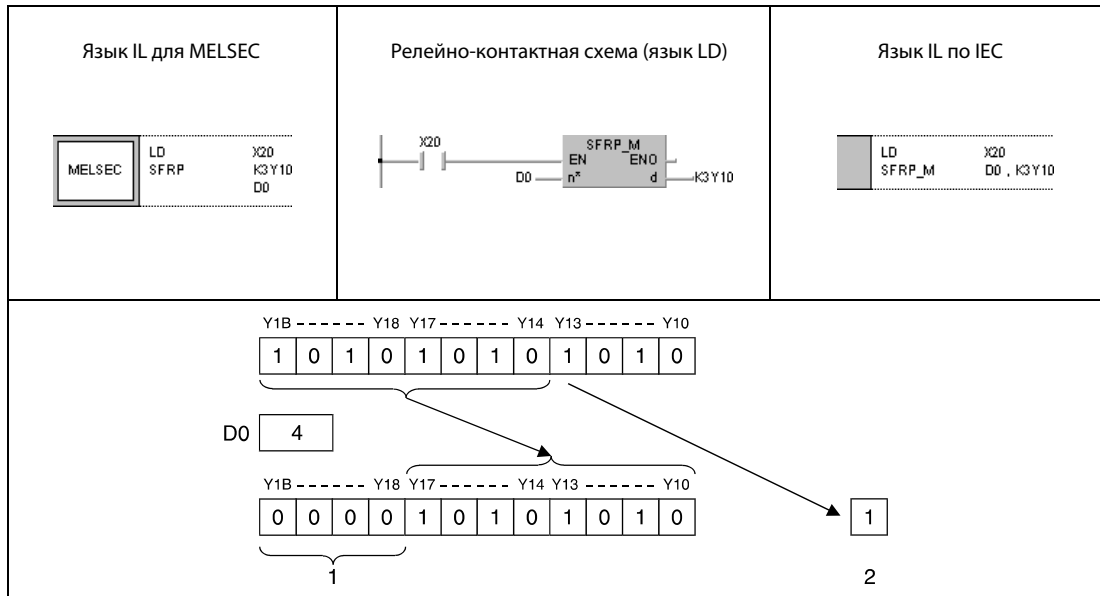
Младшие n битов, начиная с бита b0, устанавливаются на 0. n-ый сдвигаемый бит (b (15 – n)) записывается во флаг переноса (серия "А" = M9012, серии "Q"/"System Q" = SM700).

В случае таймеров или счетчиков сдвигается фактическое значение (состояние счета). Сдвиг заданного значения (уставки) не возможен.

При использовании битовых операндов возможен сдвиг в пределах операнда на указанное число битов (см. пример 2).

**Пример 1** SFRP

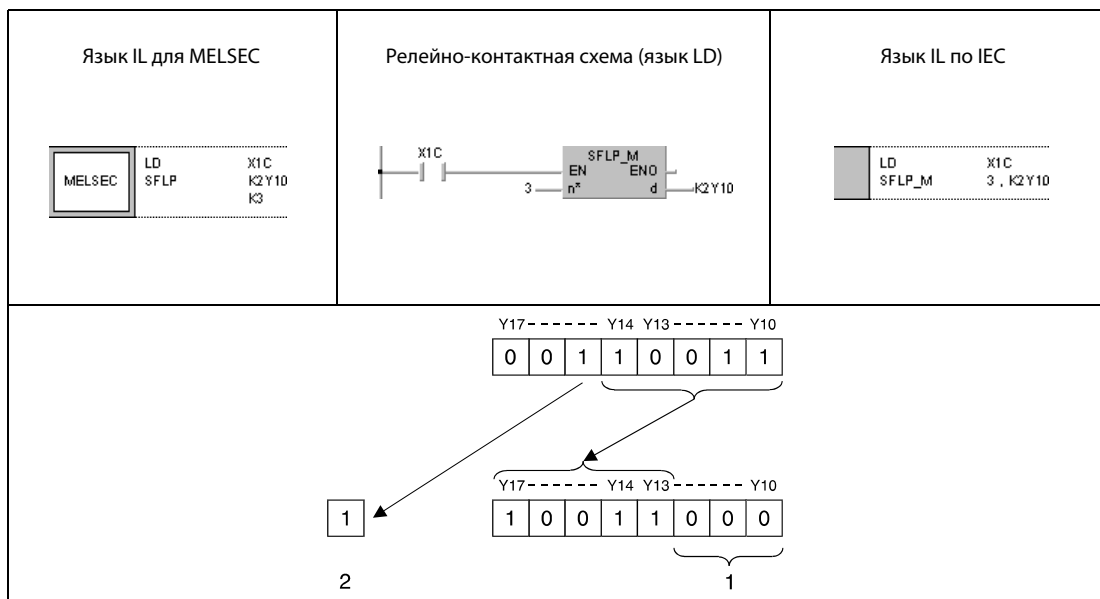
В следующей программе при положительном фронте X20 содержимое выходов Y10...Y1В сдвигается вправо на количество битов, указанное в D0. При этом состояние бита Y13 сохраняется во флаге переноса (серия "A" = M9012, серии "Q"/"System Q" = SM700).



- <sup>1</sup> В эти биты записываются нули.
- <sup>2</sup> флаг переноса

**Пример 2** SFLP

Следующая программа при положительном фронте сигнала X1C сдвигает содержимое выходов Y10...Y18 на 3 бита влево. При этом состояние Y15 сохраняется во флаге переноса (серия "A" = M9012, серии "Q"/"System Q" = SM700).



- <sup>1</sup> В эти биты записываются нули.
- <sup>2</sup> флаг переноса

**7.3.2 BSFR, BSFRP, BSFL, BSFLP**

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

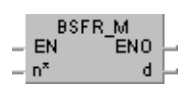
	Операнды																	Длина блока	Шагов	Индекс	Флаг пере-носа	Флаг оши-бки					
	битовые						словные (16 бит)						Константы		Указа-тели		Уро-вень										
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K						H (16#)	P	I	N	
d	●	●	●	●	●	●																	7	1	●	●	●
n																●	●							●	●	●	●

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU” этого руководства.

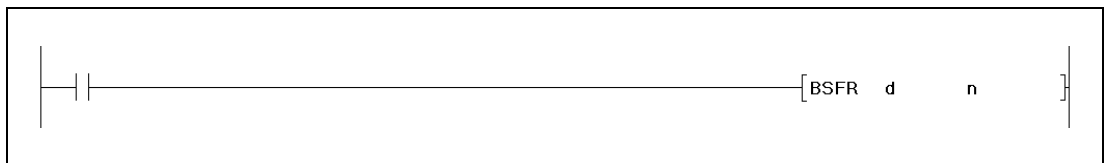
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., польз.о.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы E	Иные U		
	битовые	словные		битовые	словные						
d	●	—	—	—	—	—	—	—	—	—	—
n	—	●	●	●	●	●	●	●	—	SMO	3

**GX IEC Developer**

<p>Язык IL для MELSEC</p> <div style="border: 1px solid gray; padding: 5px; display: inline-block; margin-bottom: 5px;"> <span style="border: 1px solid gray; padding: 2px;">MELSEC</span> BSFR d n         </div>	<p>Релейно-контактная схема (язык LD)</p> 	<p>Язык IL по IEC</p> <div style="border: 1px solid gray; padding: 5px; display: inline-block; margin-bottom: 5px;"> <span style="border: 1px solid gray; padding: 2px;">BSFR_M</span> n , d         </div>
--	---	---

**GX Developer**

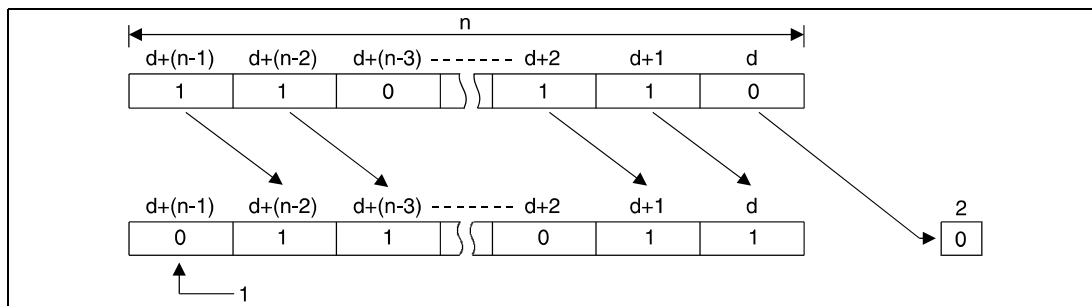


**Переменные**

Операнд	Значение	Тип данных
d	Первый адрес сдвигаемого операнда	бит
n	Количество сдвигаемых операндов	BIN, 16 бит

**Принцип действия****Сдвиг n битовых операндов на 1 бит****BSFR Сдвиг вправо**

Команда BSFR сдвигает состояния определенных битовых операндов на один бит вправо. Процесс сдвига начинается с адреса операнда, заданного в d, и выполняется для следующих n адресов.

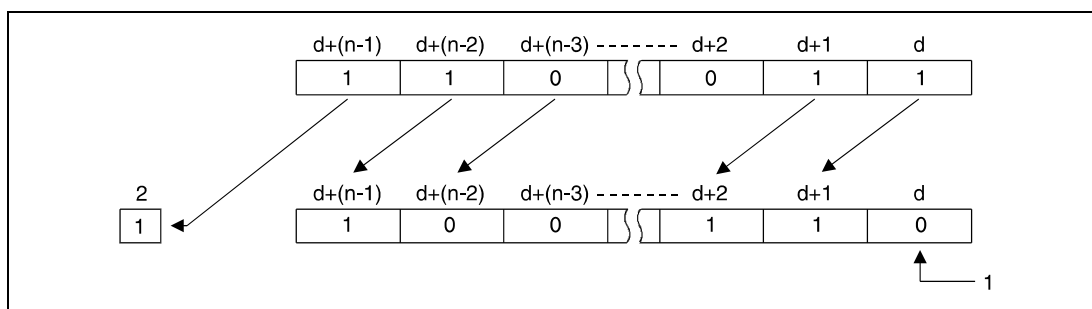


<sup>1</sup> В этот бит записывается 0.

<sup>2</sup> флаг переноса

**BSFL Сдвиг влево**

Команда BSFL сдвигает состояния определенных битовых операндов на один бит влево. Процесс сдвига начинается с адреса операнда, заданного в d, и выполняется для следующих n адресов.



<sup>1</sup> В этот бит записывается 0.

<sup>2</sup> флаг переноса

**Источники ошибок**

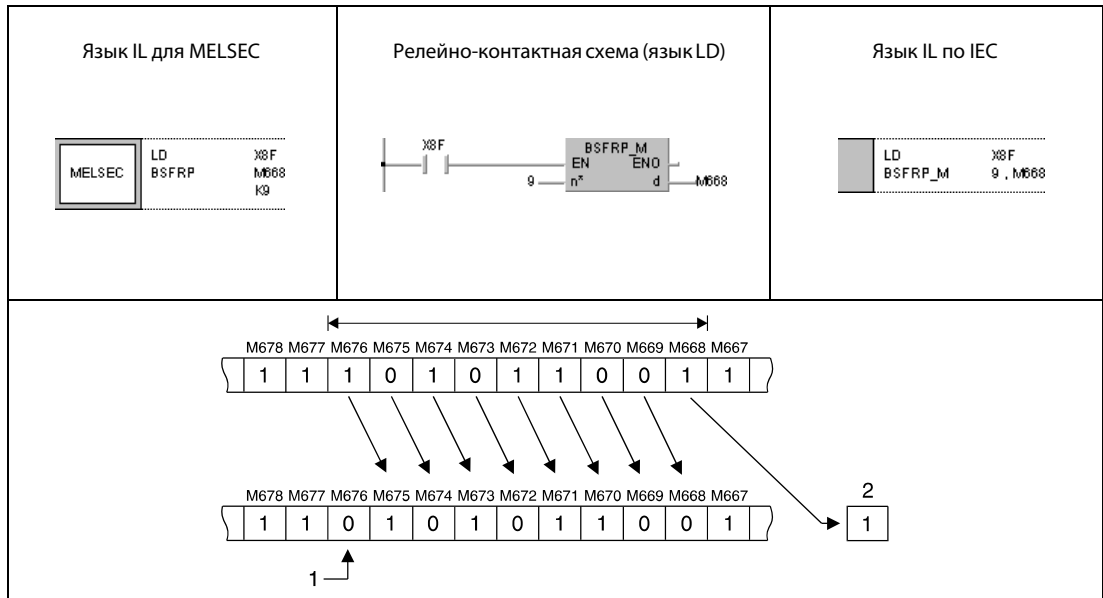
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В n указано отрицательное значение.
- Значение в n превышает число битов операнда, указанного в d (серии "Q"/"System Q" = код ошибки 4101).

**Пример 1** BSFRP

Следующая программа при положительном фронте сигнала X8F сдвигает данные маркеров M668...M676 на один бит вправо. Маркер M668 принимает значение маркера M669, маркер M669 – маркера M670 и т. д.

Состояние первого операнда (M668) записывается во флаг переноса (серия "A" = M9012, серии "Q"/"System Q" = SM700), а последнему операнду (M676) присваивается значение 0.

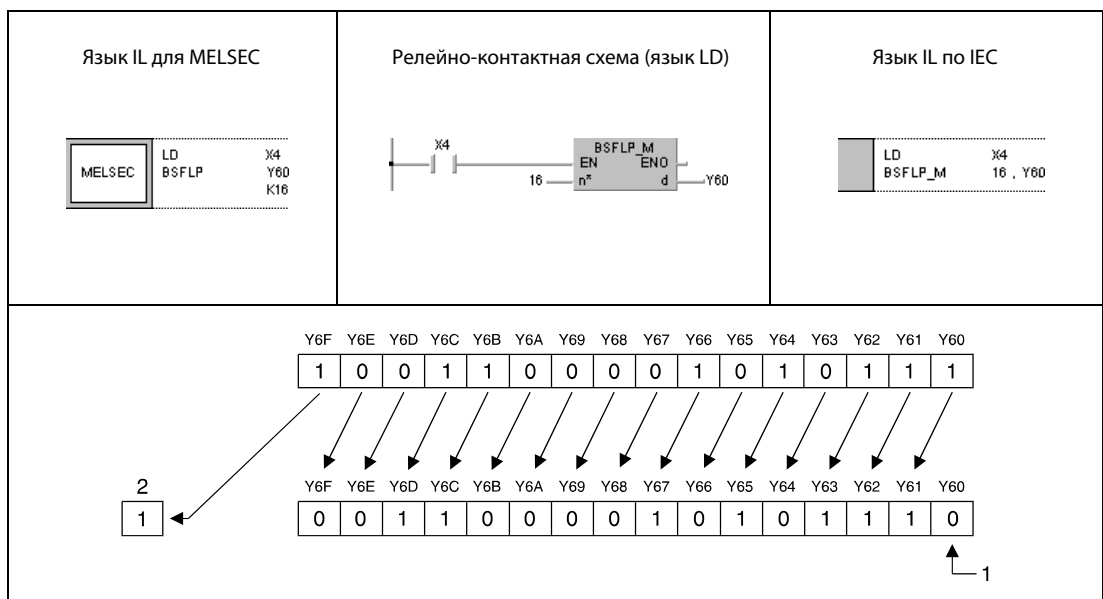


<sup>1</sup> В этот бит записывается 0.

<sup>2</sup> флаг переноса

**Пример 2** BSFLP

В следующей программе при положительном фронте сигнала X4 состояния выходов Y60...Y6F сдвигаются на один операнд влево. Состояние последнего выхода (Y6F) сохраняется во флаге переноса (серия "A" = M9012, серии "Q"/"System Q" = SM700), а первый выход (Y60) устанавливается на 0.



<sup>1</sup> В этот бит записывается 0.

<sup>2</sup> флаг переноса

### 7.3.3 DSFR, DSFRP, DSFL, DSFLP

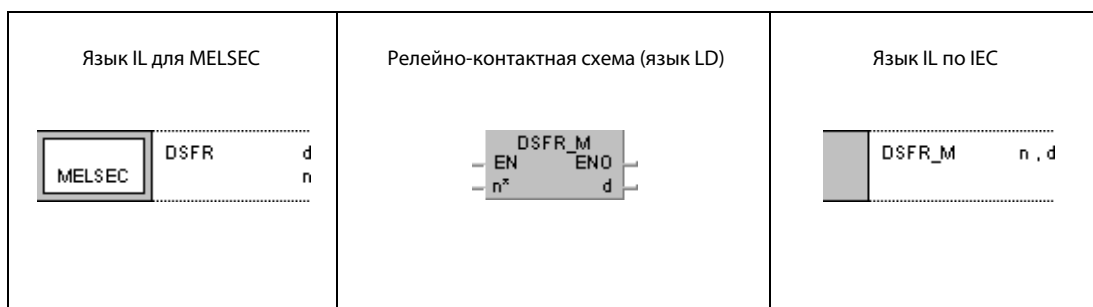
Процессор	AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
	●	●	●	●	●	●

	Операнды																Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки						
	битовые						словные (16 бит)						Константы		Указатели							Уровень					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						K	H (16#)	P	I	N	N
d							●	●	●	●	●													7 <sup>1</sup>	●		●
n																	●	●						●	●		●

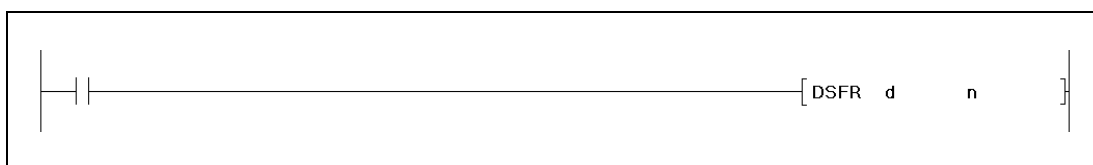
<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□VG□	Индексные регистры Zn	Константы E	Иные		
	битовые	словные		битовые	словные						
d	—	●	●	—	—	—	—	—	—	—	—
n	●	●	●	●	●	●	●	●	—	SM0	3

#### GX IEC Developer



#### GX Developer



Переменные	Операнд	Значение	Тип данных
	d	Первый адрес сдвигаемого операнда	BIN, 16 бит
	n	Количество сдвигаемых операндов	BIN, 16 бит

**Принцип действия**

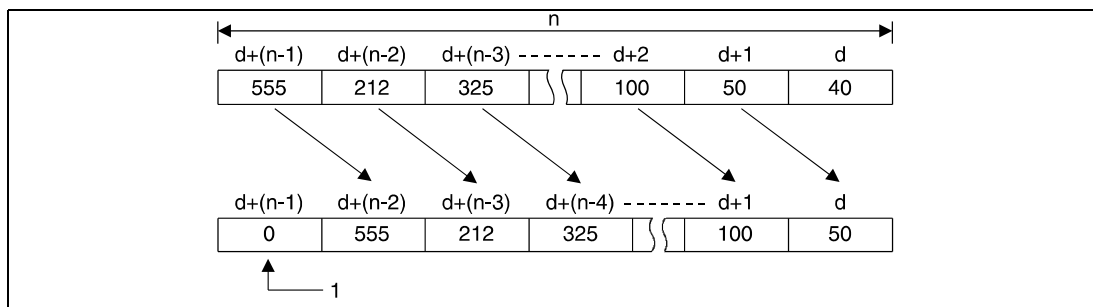
**Сдвиг n словных операндов на 1 адрес**

**DSFR Сдвиг вправо**

Команда DSFR сдвигает содержимое определенных словных операндов на один адрес вправо. Процесс сдвига начинается с адреса, заданного в d, и выполняется для следующих n адресов.

После сдвига содержимое самого старшего операнда устанавливается на 0.

В случае таймеров или счетчиков сдвигается фактическое значение (состояние счета). Сдвиг заданного значения (уставки) не возможен.



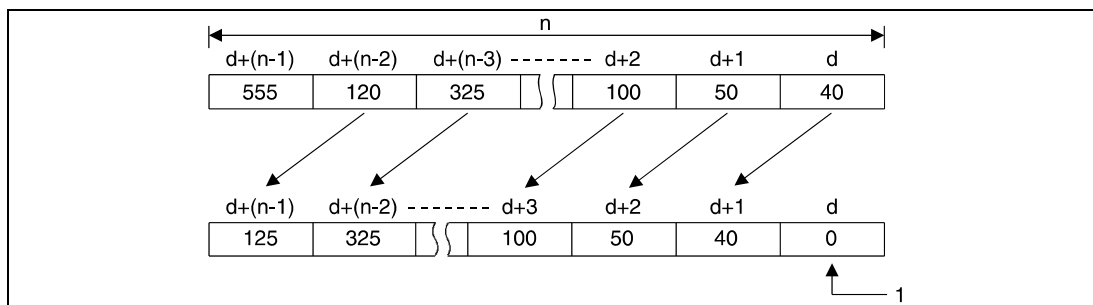
<sup>1</sup> В этот бит записывается 0.

**DSFL Сдвиг влево**

Команда DSFL сдвигает содержимое определенных словных операндов на один адрес влево. Процесс сдвига начинается с адреса, заданного в d, и выполняется для следующих n адресов.

После сдвига содержимое самого младшего операнда устанавливается на 0.

В случае таймеров или счетчиков сдвигается фактическое значение (состояние счета). Сдвиг заданного значения (уставки) не возможен.



<sup>1</sup> В этот бит записывается 0

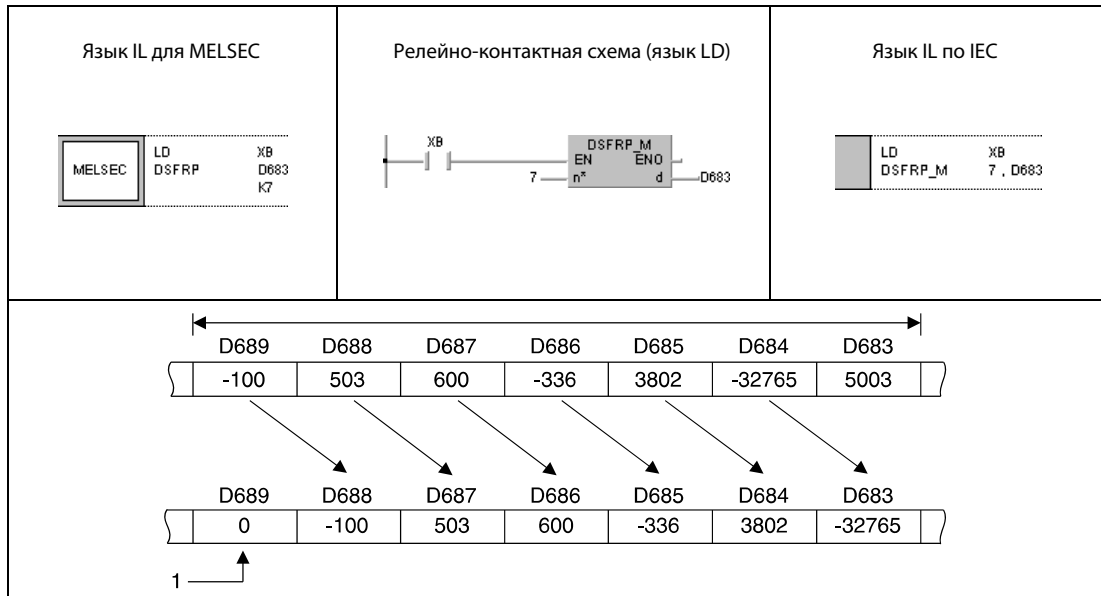
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В n указано отрицательное значение.
- Значение в n превышает число битов операнда, указанного в d (серии "Q"/"System Q" = код ошибки 4101).

**Пример 1** DSFRP

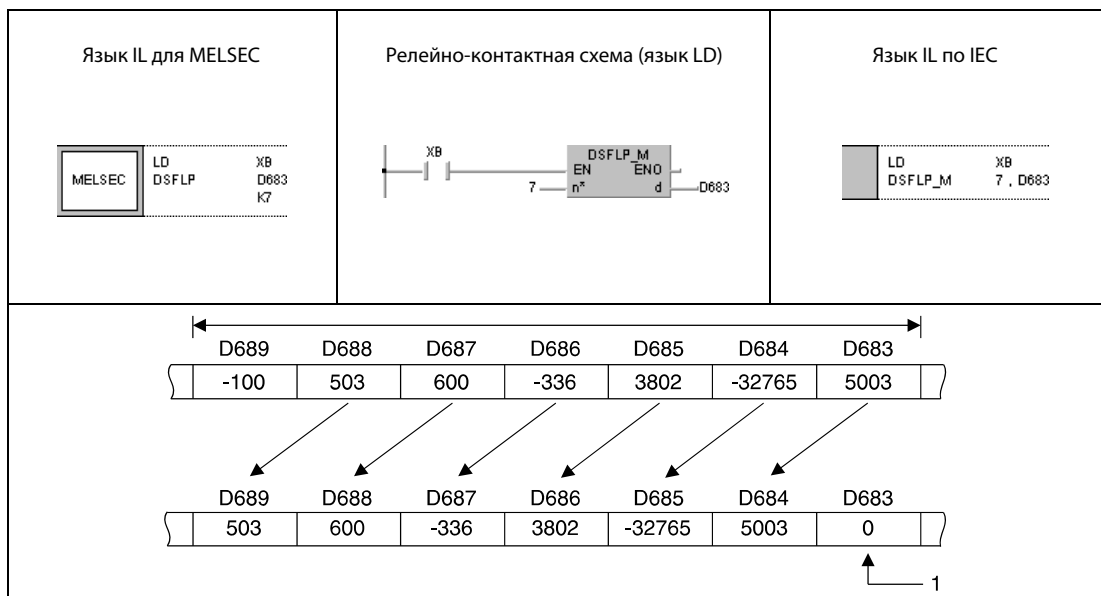
Следующая программа при положительном фронте сигнала XВ сдвигает данные регистров D683...D689 на один адрес вправо. D683 принимает значение D684, D684 принимает значение D685 и т. д. Последнему регистру данных (D689) присваивается значение 0.



<sup>1</sup> В этот бит записывается 0.

**Пример 2** DSFLP

Следующая программа при положительном фронте XВ сдвигает данные регистров D683... D689 на один адрес влево. D689 принимает значение D688, D688 принимает значение D687 и т. д. Первому регистру данных (D683) присваивается значение 0.



<sup>1</sup> В этот бит записывается 0.



## 7.4 Команды обработки битов

Команды обработки битов позволяют изменять состояния (установка и сброс) отдельных битов или областей битов. С помощью команд обработки битов можно также опрашивать состояния битов в словах данных.

В общей сложности имеются 10 команд обработки битов.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе IEC
Установка и сброс отдельных битов	BSET	BSET_M
	BSETP	BSETP_M
	BRST	BRST_M
	BRSTP	BRSTP_M
Опрос состояния отдельных битов в 16/32-битных словах данных	TEST	TEST_M
	TESTP	TESTP_M
	DTEST	DTEST_M
	DTESTP	DTESTP_M
Сброс областей битов	BKRST	BKRST_M
	BKRSTP	BKRSTP_M

### 7.4.1 BSET, BSETP, BRST, BRSTP

Процессор	AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
	●	●	●	●	●	●

**Операнды MELSEC A**

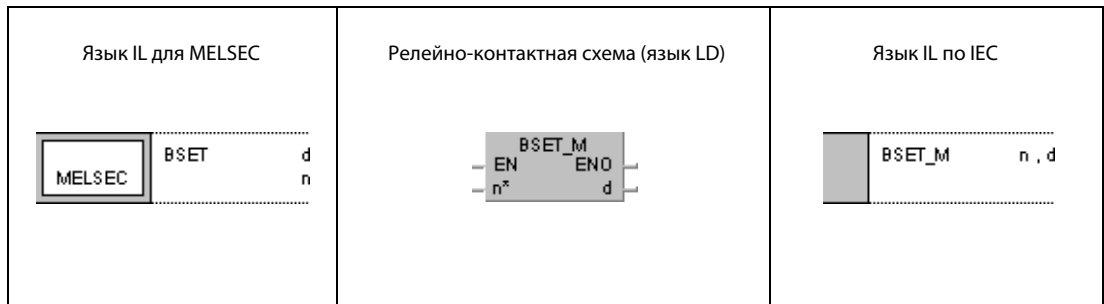
	Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9011	
	битовые						словные (16 бит)						Константы		Указатели	Уровень						
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						K
d							●	●	●	●	●	●	●	●								
n																●	●					

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

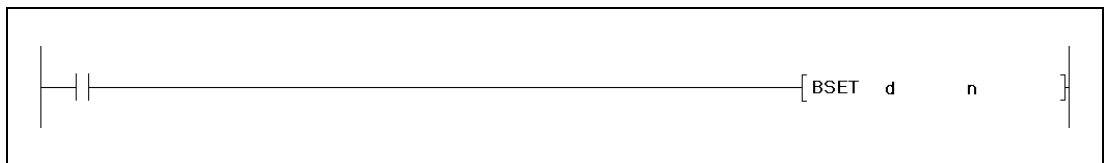
**Операнды MELSEC Q**

	Операнды										Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы K, H (16#)	Иные			
	битовые	словные		битовые	словные							
d	●	●	●	●	●	●	●	—	—	—	3	
n	●	●	●	●	●	●	●	●	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

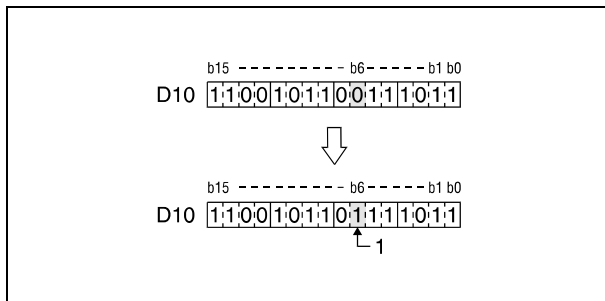
Операнд	Значение	Тип данных
d	Операнд, в котором устанавливаются или сбрасываются биты.	BIN, 16 бит
n	Адрес устанавливаемого или сбрасываемого бита	

**Принцип действия**

**Установка и сброс отдельных битов**

**BSET Установка отдельных битов в словном операнде**

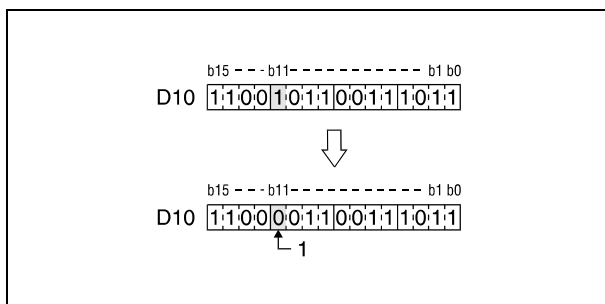
Команда BSET устанавливает n-ый бит словного операнда на 1. Для n можно использовать значение между 0 и 15 (соответственно, b0 и b15). Словный операнд устанавливается в d. Если значение в n больше 15, команда BSET выполняется в пределах первых 4 битов (от b0 до b3) словного операнда. На следующей иллюстрации указано n = 6 и устанавливается бит b6.



<sup>1</sup> Устанавливается этот бит.

**BRST Сброс отдельных битов в словном операнде**

Команда BRST сбрасывает n-ый бит словного операнда на 0. Для n можно использовать значение между 0 и 15 (соответственно, b0 и b15). Словный операнд устанавливается в d. Если значение в n больше 15, команда BRST выполняется в пределах первых 4 битов (от b0 до b3) словного операнда. На следующей иллюстрации указано n = 11 и сбрасывается бит b11.

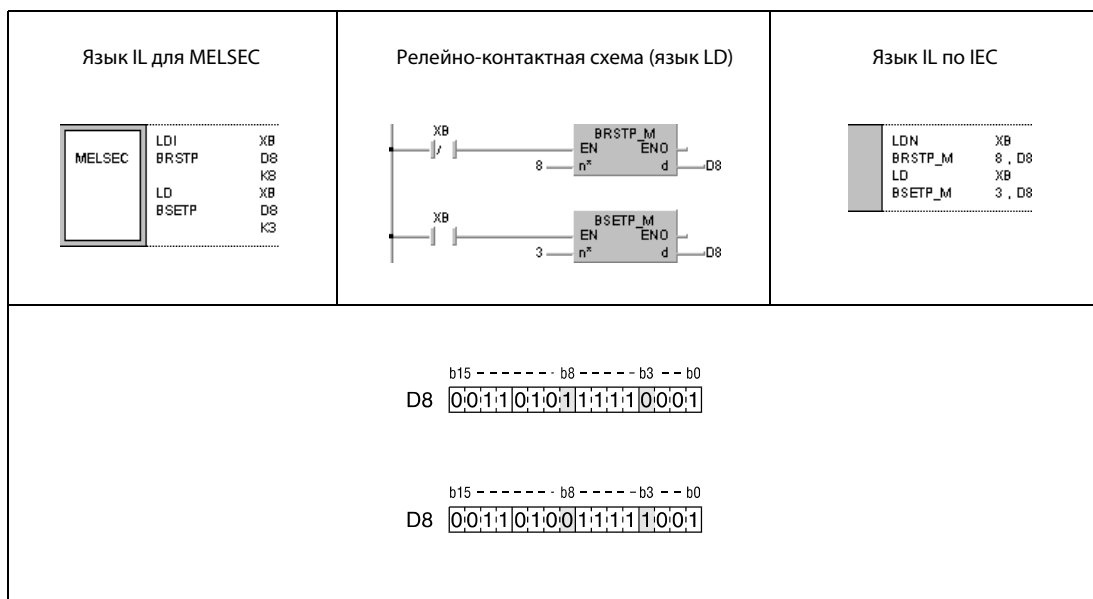


<sup>1</sup> Сбрасывается этот бит.

**Пример**

**BRSTP/BSETP**

Следующая программа при положительном фронте сигнала XВ устанавливает бит b3 регистра D8 на 1. При положительном фронте размыкающего контакта XВ бит b8 сбрасывается на 0.



**ПРИМЕЧАНИЕ**

Для установки и сброса отдельных битов в словном операнде можно также использовать команды SET и RST. В этом случае биты в словах данных указываются при адресации регистров. Например, бит b8 в слове данных D5 адресуется как D5.8.

### 7.4.2 TEST, TESTP, DTEST, DTESTP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

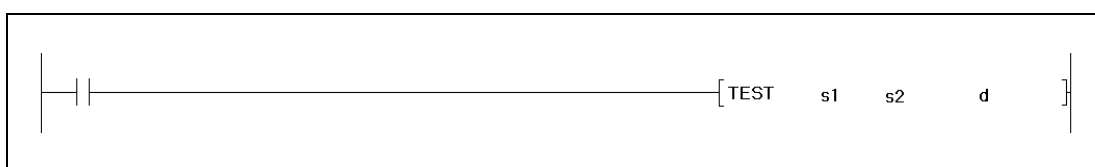
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	—	—	—	—	—	—	—	—	4	
s2	—	—	—	—	—	—	—	—	—		
d	●	—	—	●	—	—	—	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s1	Операнд, в котором находятся опрашиваемые биты.	слово
s2	Указание опрашиваемого бита	слово
d	Битовый операнд, в котором сохраняется результат опроса.	бит

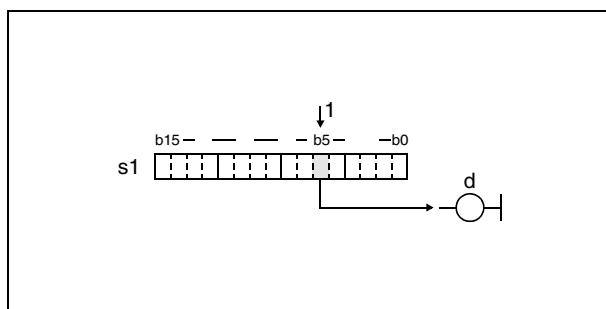
**Принцип действия****Опрос состояния отдельных битов в 16/32-битных словах данных****TEST Команда опроса, 16-битная**

Команда TEST опрашивает состояние бита  $s2$  в словном операнде  $s1$ . Результат опроса сохраняется в битовом операнде, указанном в  $d$ .

Если опрошенный бит имеет состояние 1, указанный в  $d$  битовый операнд устанавливается. Если опрошенный бит имеет состояние 0, указанный в  $d$  битовый операнд сбрасывается.

В  $s2$  может быть указан любой бит (от  $b0$  до  $b15$ ) в 16-битном слове данных. Если в  $s2$  указано значение больше 15, то опрашивается бит, номер которого является остатком от деления содержимого  $s2$  на 16. Например, если содержимое  $s2 = 18$ , опрашивается  $b2$  (при делении 18 на 16 образуется остаток 2).

В следующем примере указано  $s2 = 5$  и опрашивается состояние бита  $b5$  в  $s1$ .



<sup>1</sup> опрошенный бит

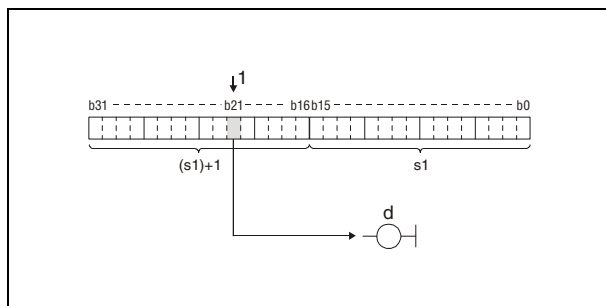
**DTEST Команда опроса, 32-битная**

Команда DTEST опрашивает состояние бита  $s2$  в словных операндах  $s1$  и  $(s1) + 1$ . Результат опроса сохраняется в битовом операнде, указанном в  $d$ .

Если опрошенный бит имеет состояние 1, указанный в  $d$  битовый операнд устанавливается. Если опрошенный бит имеет состояние 0, указанный в  $d$  битовый операнд сбрасывается.

В  $s2$  может быть указан любой бит (от  $b0$  до  $b31$ ) 32-битного слова данных. Если в  $s2$  указано значение больше 31, то опрашивается бит, номер которого равен остатку от деления содержимого  $s2$  на 32. Например, если  $s2 = 34$ , опрашивается бит  $b2$  (так как при делении 34 на 32 остаток равен 2).

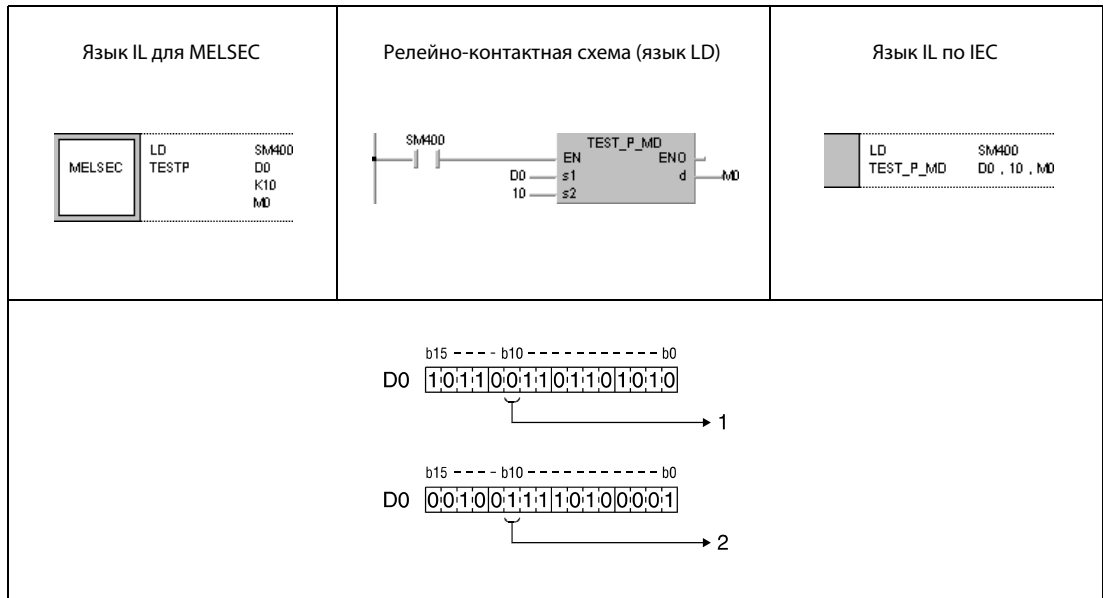
В следующем примере указано  $s2 = 21$  и опрашивается состояние бита  $b21$  в  $s1$ .



<sup>1</sup> опрошенный бит

**Пример 1** TESTP

Следующая программа при положительном фронте сигнала SM400 в зависимости от результата опроса бита b10 в 16-битном слове данных D0 сбрасывает или устанавливает маркер M0.

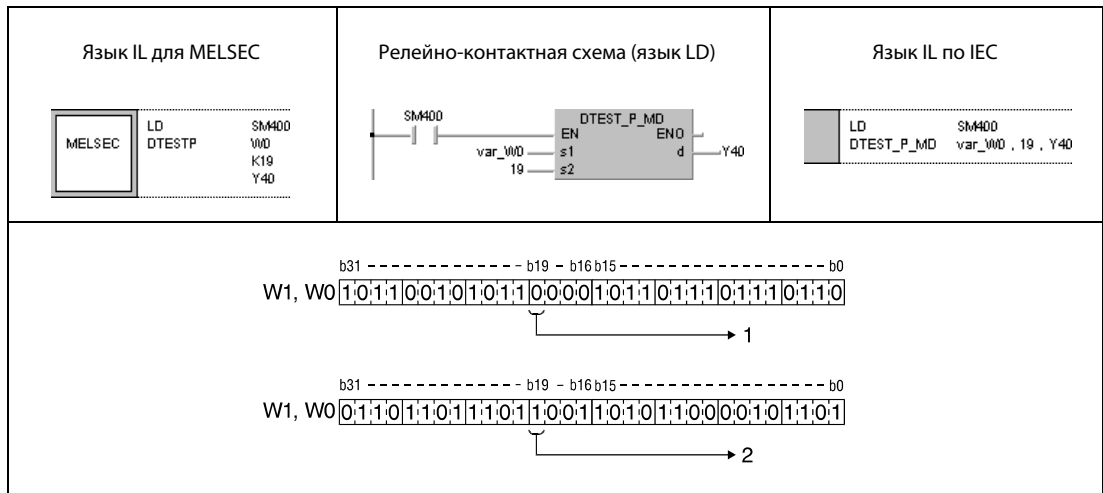


1 = сброс

2 = установка

**Пример 2** DTESTP

Следующая программа при положительном фронте сигнала SM400 в зависимости от результата опроса бита b19 в 32-битном слове данных в W0 и W1 сбрасывает или устанавливает выход Y40.

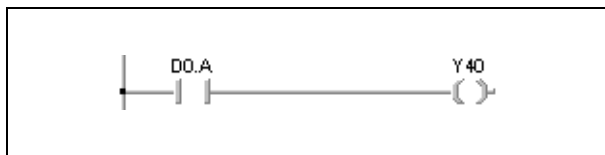


1 = сброс

2 = установка

**ПРИМЕЧАНИЯ** Без объявления переменной в заголовке программного компонента (POU) пример программы 2 не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

Вместо применения команды TEST можно также определить опрашиваемый бит в качестве входного контакта (см. рис.).





### 7.4.3 BKRST, BKRSTP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

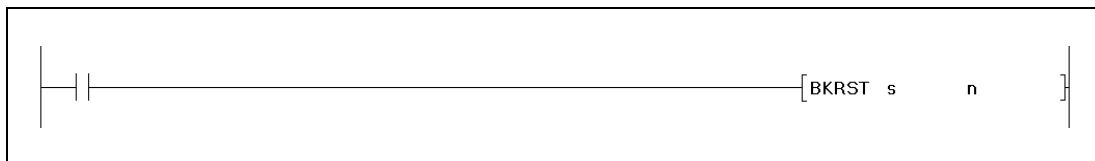
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	●	●	●	—	—	—	—	—	SM0	3	
n	●	●	●	●	●	●	●	—			

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес сбрасываемых операндов	бит
n	Количество сбрасываемых операндов	BIN, 16 бит

**Принцип действия**

**Сброс областей битов**  
**BKRST Команда сброса**

Команда BKRST сбрасывает n битов операнда, указанного в s.

Если сбрасываемыми операндами являются маркеры ошибок (F), то сбрасывается указанное в n количество маркеров ошибок, хранящихся в s, и в соответствии со сброшенными маркерами стирается содержимое регистров с SD64 по SD79. Остающиеся данные сдвигаются вперед. Кроме того, количество маркеров ошибок, записанных в регистры SD64...SD79, сохраняется в регистре SD63.

Если сбрасываемыми операндами являются таймеры (T) и счетчики (C), то в результате выполнения команды настройки n таймеров или счетчиков устанавливаются на ноль, а контакты катушек сбрасываются.

В случае всех прочих битовых операндов указанное n количество катушек или контактов сбрасывается в операнде, указанном в s.

Если соответствующий операнд уже сброшен, то в результате выполнения этой команды его состояние не изменяется.

**Источники ошибок**

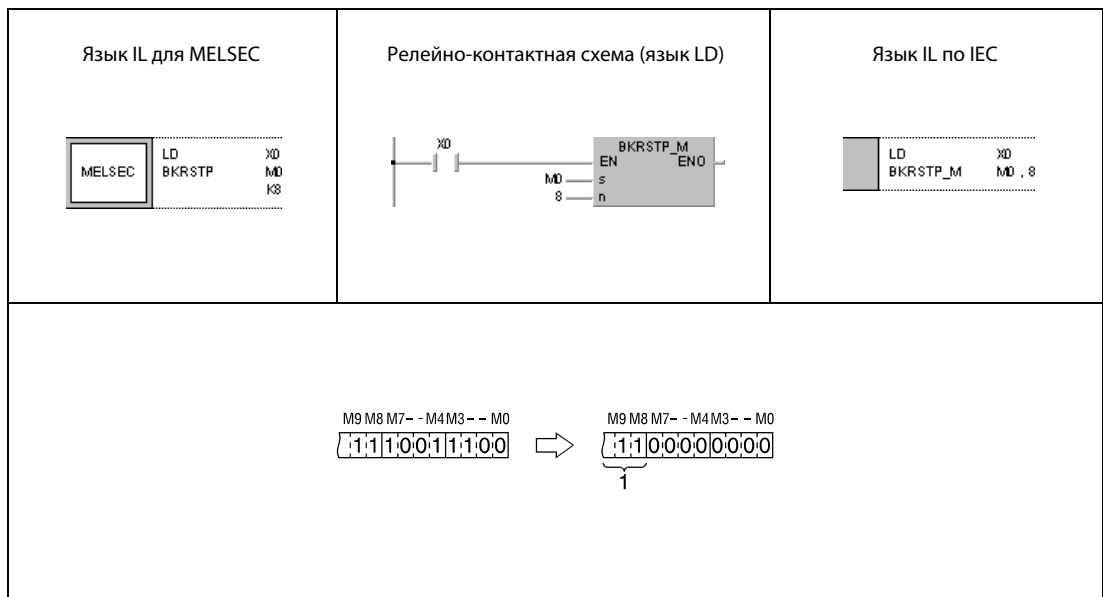
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Значение в n превышает число битов операнда, указанного в s (код ошибки 4101).

**Пример 1**

BKRSTP

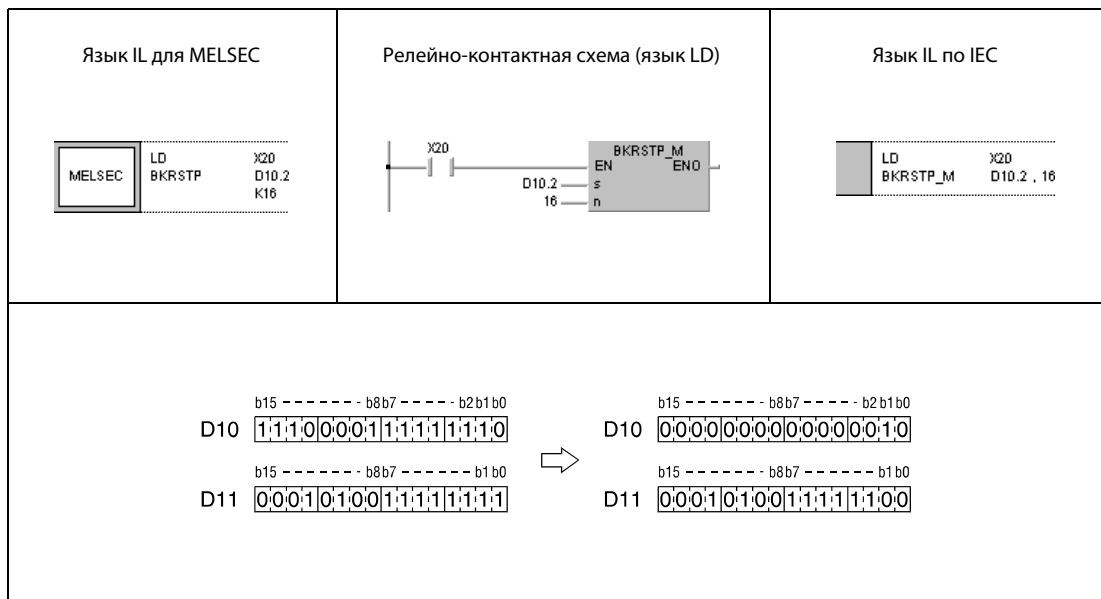
Следующая программа при положительном фронте сигнала X0 сбрасывает маркеры с M0 по M7.



<sup>1</sup> Эти биты не изменяют свое состояние.

**Пример 2** BKRSTP

Следующая программа при положительном фронте сигнала X20 сбрасывает биты от бита b2 в D10 до бита b1 в D11.





## 7.5 Команды обработки данных

Команды обработки данных позволяют искать данные в определенных областях, определять количество установленных битов, кодировать и декодировать данные (в том числе для 7-сегментной индикации), разделять или объединять значения данных, искать максимальные и минимальные значения, сортировать данные и суммировать блоки 16/32-битных двоичных данных.

В общей сложности имеется 41 команда обработки данных. Следующая таблица содержит обзор этих команд.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту IEC
Поиск 16/32-битных данных	SER	SER_M
	SERP	SERP_M
	DSER	DSER_M
	DSERP	DSERP_M
Контроль битов данных (16/32 бита)	SUM	SUM_M
	SUMP	SUMP_M
	DSUM	DSUM_M
	DSUMP	DSUMP_M
Кодировка и декодировка данных	DECO	DECO_M
	DECOP	DECOP_M
	ENCO	ENCO_M
	ENCOP	ENCOP_M
7-сегментная декодировка	SEG	SEG_M
Разъединение и объединение 16-битных слов данных (4-битные единицы)	DIS	DIS_M
	DISP	DISP_M
	UNI	UNI_M
	UNIP	UNIP_M
Разъединение и объединение 16-битных значений данных (единицы с переменным числом битов)	NDIS	NDIS_M
	NDISP	NDISP_M
	NUNI	NUNI_M
	NUNIP	NUNIP_M
Разъединение и объединение 16-битных значений данных (байтовые единицы)	WTOB	WTOB_MD
		WTOB_K_MD
	WTOBP	WTOB_P_MD
		WTOB_K_P_MD
	BTOW	BTOW_MD
		BTOW_K_MD
BTOWP	BTOW_P_MD	
	BTOW_K_P_MD	
Поиск максимальных значений в 16/32-битных данных	MAX	MAX_M
	MAXP	MAXP_M
	DMAX	DMAX_M
	DMAXP	DMAXP_M

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту IEC
Поиск минимальных значений в 16/32-битных данных	MIN	MIN_M
	MINP	MINP_M
	DMIN	DMIN_M
	DMINP	DMINP_M
Сортировка 16/32-битных данных	SORT	SORT_M
	SORTP	SORTP_M
	DSORT	DSORT_M
	DSORTP	DSORTP_M
Вычисление суммы блоков 16/32-битных двоичных данных	WSUM	WSUM_M
	WSUMP	WSUMP_M
	DWSUM	DWSUM_M
	DWSUMP	DWSUMP_M

### 7.5.1 SER, SERP, DSER, DSERP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

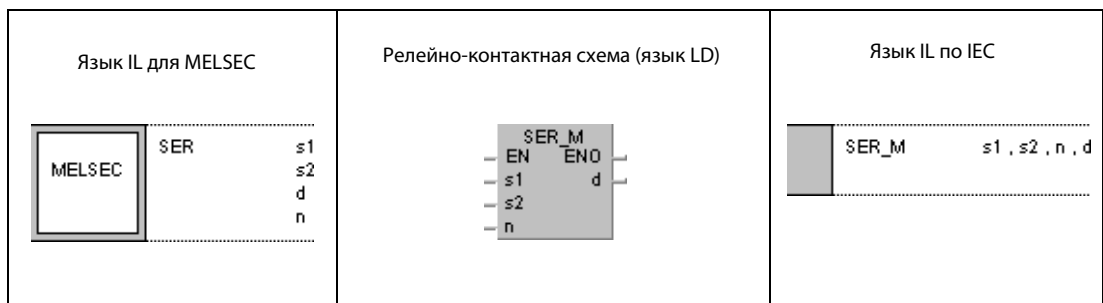
	Операнды															Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011	
	битовые					словные (16 бит)					Константы		Указатели		Уровень						
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V
s1								●	●	●	●	●	●	●	●	●	●	●			
s2								●	●	●	●	●									
n																	●	●			

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

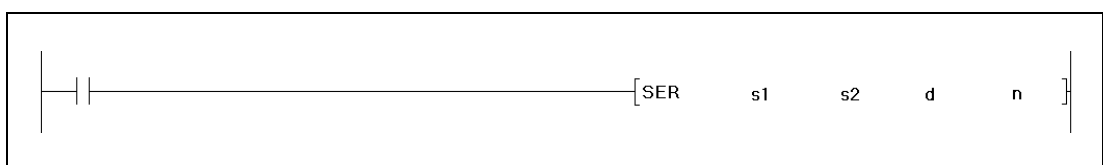
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы K, H (16#)			Иные U
	битовые	словные		битовые	словные						
s1	●	●	●	●	●	●	●	●	—	SMO	5
s2	—	●	●	—	—	—	—	—			
d	—	●	●	—	●	●	●	—			
n	●	●	●	●	●	●	●	●	—		

**GX IEC Developer**



**GX Developer**



**ПРИМЕЧАНИЕ**

В случае серии "А" результат поиска всегда сохраняется в регистрах A0 и A1. Поэтому при программировании этой команды в серии "А" отсутствует операнд d.

**Переменные**

Операнд	Значение	Тип данных	
		MELSEC	IEC
s1	Искомое значение данных или первый адрес операнда, в котором это значение сохранено.	слово	ANY16
s2	Данные, в которых ведется поиск значения, или первый адрес операнда, в котором сохранены эти данные.		ANY16/ANY32
d	Первый адрес операнда, в котором сохраняется результат поиска. В серии "A" это всегда операнды A0 и A1.		массив [1..2] данных типа ANY16 / ANY32
n	Количество проверяемых адресов		ANY16

**Принцип действия**

**Поиск данных**

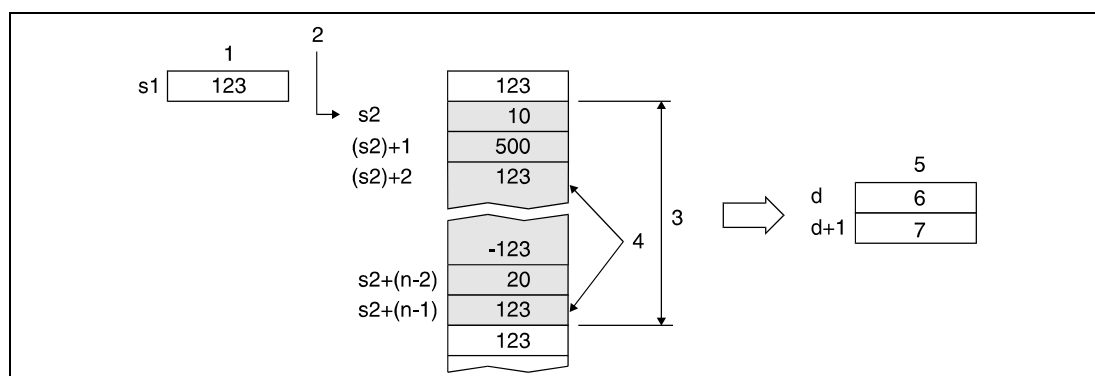
**SER (серии "A" и "Q"/"System Q")/SERP (серии "Q"/"System Q") Поиск 16-битных данных**

Команда SER позволяет искать определенные данные в определенной области. Процесс поиска начинается с указанного в s2 начального адреса. Искомое значение задается в s1. Длина области поиска (т. е. число адресов) устанавливается в n.

В процессорах серий "Q" и "System Q" результат поиска сохраняется в (Array\_d[1]) и (Array\_d[2]).

После поиска позиция первого адреса, в которой расположено искомое значение, сохраняется в массиве (Array\_d[1]). Массив (Array\_d[2]) содержит количество найденных данных, идентичных искомым данным.

В случае серии "A" позиция первого адреса найденного значения сохраняется в регистре A0. Количество найденных данных записывается в регистр A1.



- 1 искомое значение
- 2 начальный адрес поиска
- 3 область поиска
- 4 совпадающие значения
- 5 результат поиска
- 6 адрес первого найденного значения
- 7 количество найденных значений

Если значение в n отрицательное или равно 0, поиск данных не выполняется. Если при поиске не найдены совпадающие данные, в массивы (Array\_d[1]) и (Array\_d[2]) или в регистры A0 и A1 записывается "0".

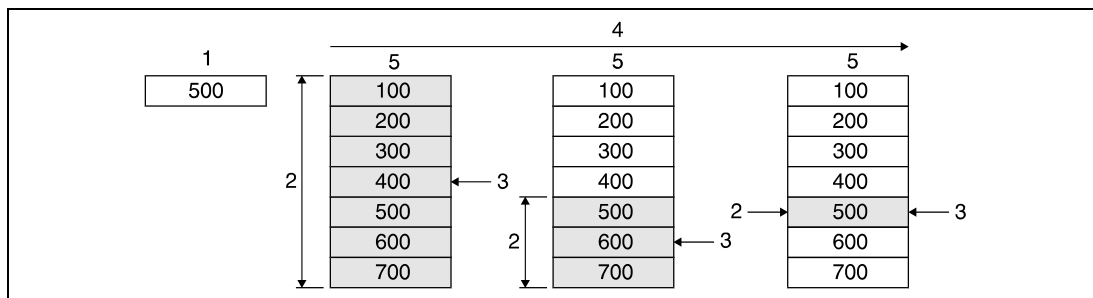


**ПРИМЕЧАНИЯ** Серии "Q" и "System Q"

Если проверяемые данные упорядочены в возрастающей последовательности, время обработки можно сократить, применив двоичный поиск при включенном маркере диагностики SM702.

SM702 включен:

Проверяемая область уменьшается вдвое и по величине искомого значения определяется, в какой половине может находиться искомое значение. Затем выбирается эта половина области поиска и она снова уменьшается вдвое, чтобы определить очередную половину для дальнейшего поиска. Этот процесс продолжается до тех пор, пока не обнаруживается искомое значение.



- <sup>1</sup> искомое значение
- <sup>2</sup> область поиска
- <sup>3</sup> сравнение с искомым значением
- <sup>4</sup> последовательность обработки
- <sup>5</sup> Данные, в которых ведется поиск.

SM702 выключен:

Поиск ведется путем сравнения искомого значения с каждым из значений проверяемых данных. Поиск начинается с начала проверяемой области.

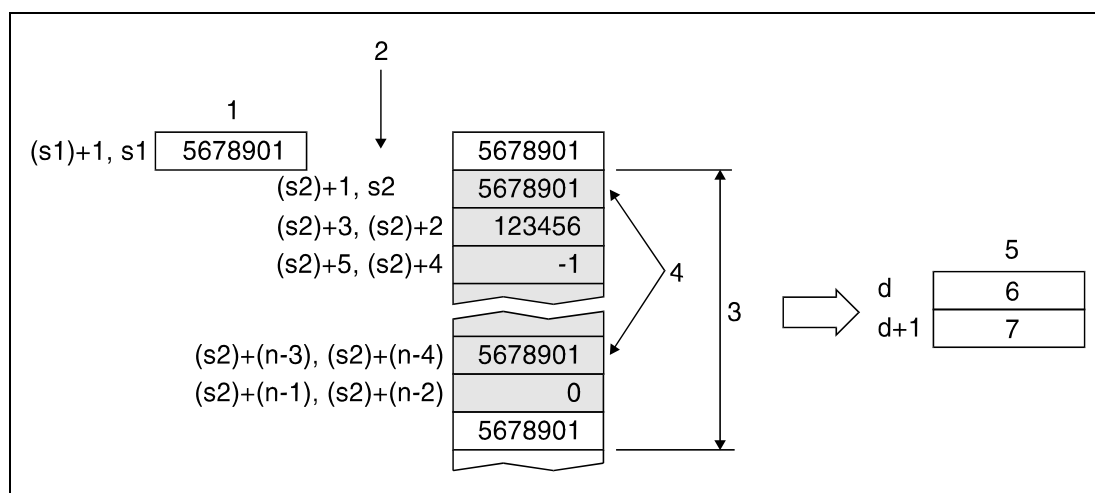
Если в проверяемой области данные не упорядочены по величине, установка маркера диагностики SM702 даст неправильный результат поиска.

**DSER / DSERP (серии "Q" и "System Q") Поиск 32-битных данных**

Команда DSER позволяет искать определенные 32-битные данные в определенной области. Поиск начинается с указанного в  $s2$  начального адреса ( $2 \times n$  адресов). Искомое значение задается в  $s1$  и  $(s1) + 1$ . Длина области поиска (т. е. число адресов) устанавливается в  $n$ .

Результат поиска сохраняется в  $d$  и  $d1$  в виде массива [1..2] данных типа ANY16.

После поиска позиция первого адреса, в которой находится искомое значение, сохраняется в  $d$  ( $Array\_d[1]$ ). Массив ( $Array\_d[2]$ ) содержит количество найденных данных, идентичных искомым данным.



- 1 искомое значение
- 2 начальный адрес поиска
- 3 область поиска
- 4 совпадающие значения
- 5 результат поиска
- 6 адрес первого найденного результата
- 7 количество найденных значений

Если значение в  $n$  отрицательное или равно 0, поиск данных не выполняется.

Если при поиске не обнаружены совпадающие данные, то в массивы ( $Array\_d[1]$ ) и ( $Array\_d[2]$ ) записывается 0.

**Источники ошибок**

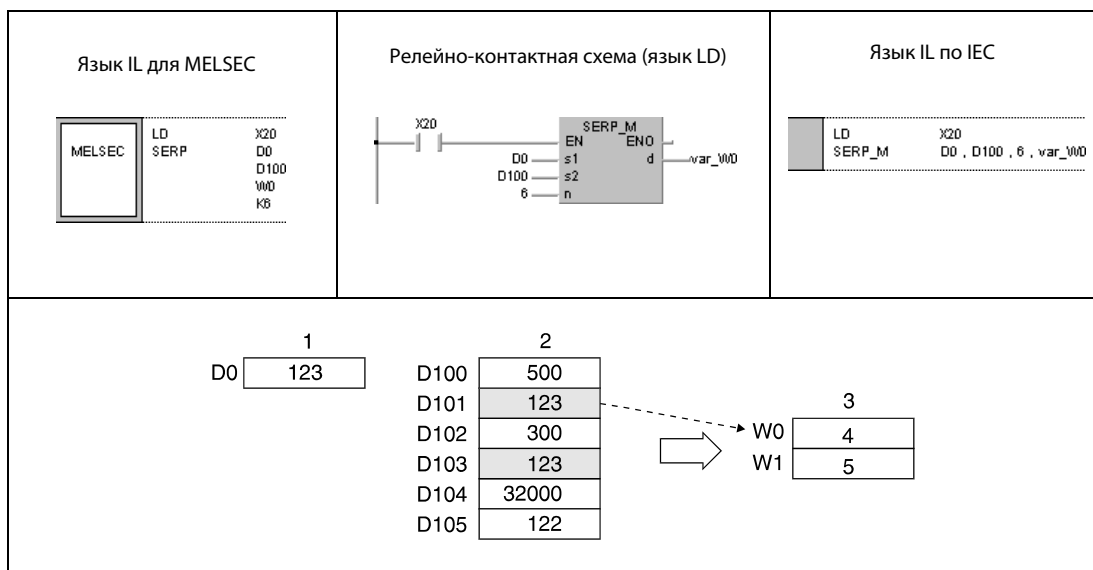
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанный с помощью  $n$  диапазон адресов, начиная с  $s2$ , находится вне области операнда, предусмотренной для сохранения (код ошибки 4101).

Информация о применении индексированной адресации имеется в разделе 3.6.

**Пример 1** SERP (серии "Q" и "System Q")

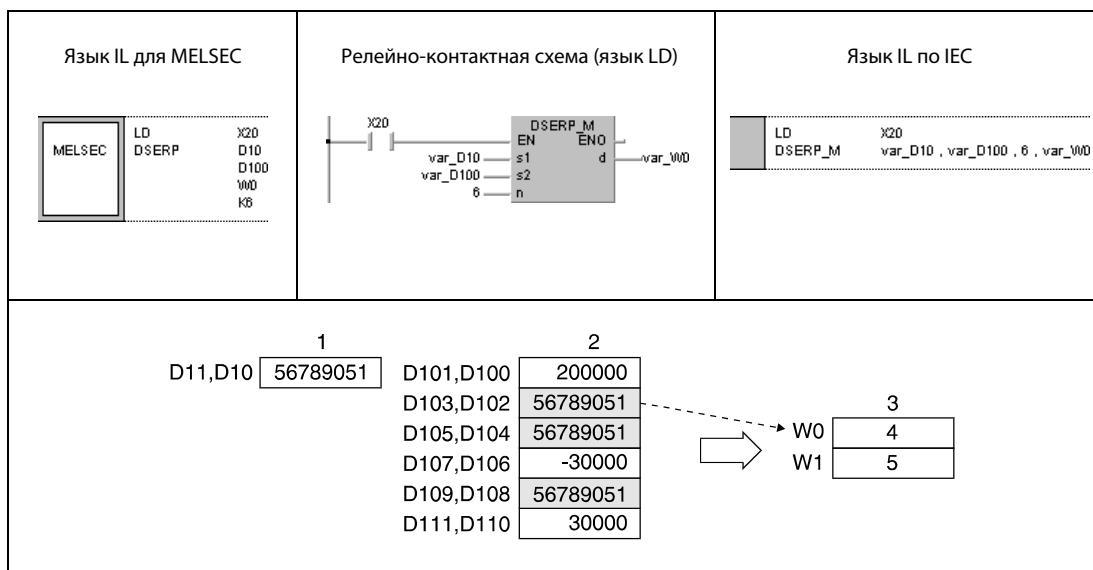
В этом примере программа при положительном фронте сигнала X20 сравнивает данные в D100...D105 со значением данных в D0. Первая совпадающая позиция сохраняется в W0, а количество позиций – в W1.



- <sup>1</sup> искомое значение
- <sup>2</sup> Данные, в которых ведется поиск.
- <sup>3</sup> результат поиска
- <sup>4</sup> адрес первого найденного результата
- <sup>5</sup> количество найденных значений

**Пример 2** DSERP (серии "Q" и "System Q")

В этом примере программа при положительном фронте сигнала X20 сравнивает данные из D100...D111 со значением в D11 и D10. Первая совпадающая позиция сохраняется в W0, а количество позиций – в W1.



- 1 искомое значение
- 2 Данные, в которых ведется поиск.
- 3 результат поиска
- 4 адрес первого найденного результата
- 5 количество найденных значений

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.5.2 SUM, SUMP, DSUM, DSUMP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

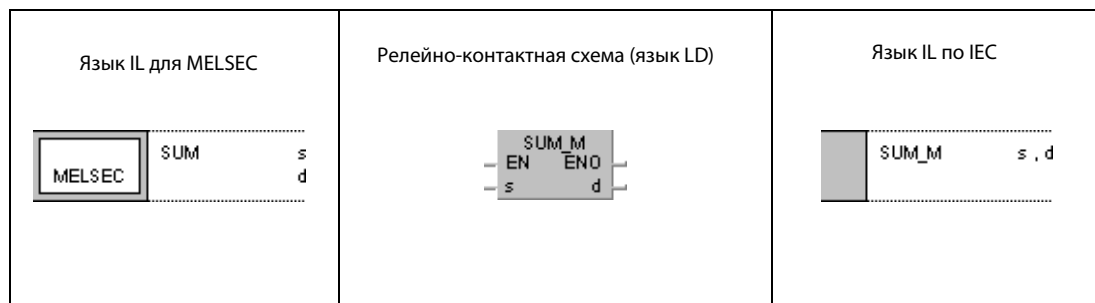
Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011							
битовые						словные (16 бит)						Константы		Указатели							Уровень						
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						K	H (16#)	P	I	N		
SUM																											
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							K1 ↓ K4	3	●		●	
DSUM																											
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●								K1 ↓ K4	3	●		●

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

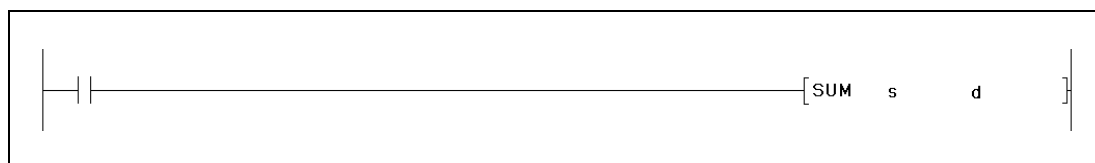
**Операнды MELSEC Q**

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непос. адр. J□□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные							
s	●	●	●	●	●	●	●	—	—	—	3
d	●	●	●	●	●	●	—	—	—	—	

**GX IEC Developer**



**GX Developer**



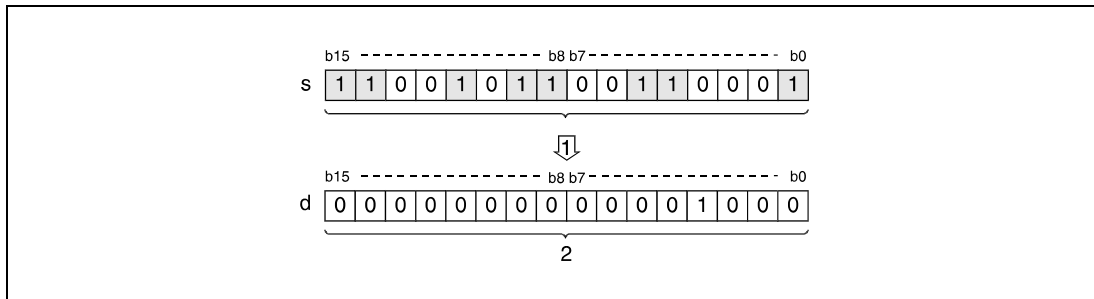
**ПРИМЕЧАНИЕ** В случае серии "A" количество установленных битов всегда сохраняется в регистре A0. Поэтому при программировании этой команды в серии "A" отсутствует операнд d.

**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором подсчитываются установленные биты.	BIN, 16/32 бита
d	Первый адрес операнда, в котором сохраняется количество установленных битов. В случае серии "A" это всегда операнд A0.	

**Принцип действия**      **Контроль битов данных**  
**SUM      16 бит**

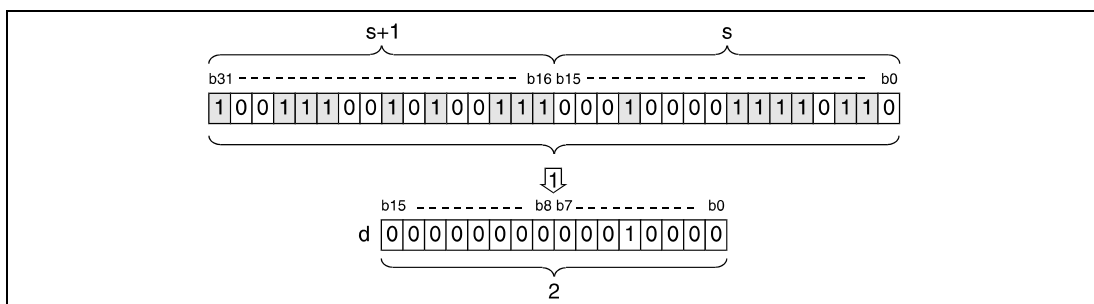
С помощью команды SUM можно определить количество битов, установленных в 16-битном слове данных 1. Проверяемый диапазон адресов задается в s. После выполнения команды сумма установленных битов записывается в d (A0).



- <sup>1</sup> счет установленных битов
- <sup>2</sup> двоичное представление количества установленных битов

**DSUM      32 бита**

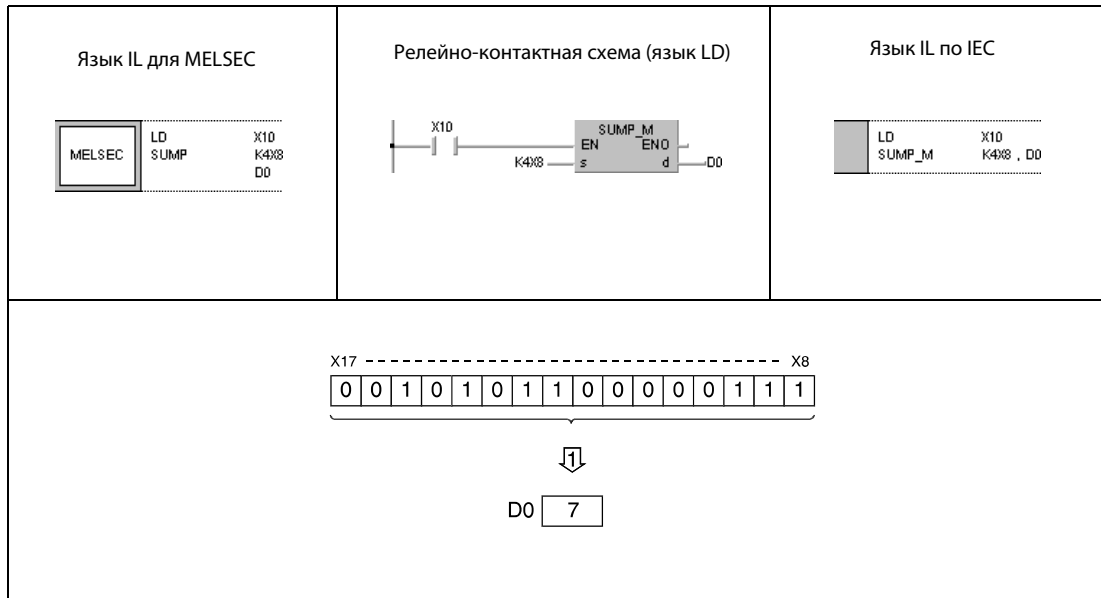
С помощью команды DSUM можно определить количество битов, установленных в 32-битном слове данных. Проверяемый диапазон адресов задается в s. После выполнения команды сумма битов со значением 1 записывается в d (A0).



- <sup>1</sup> счет установленных битов
- <sup>2</sup> двоичное представление количества установленных битов

**Пример 1** SUMP (серии "Q" и "System Q")

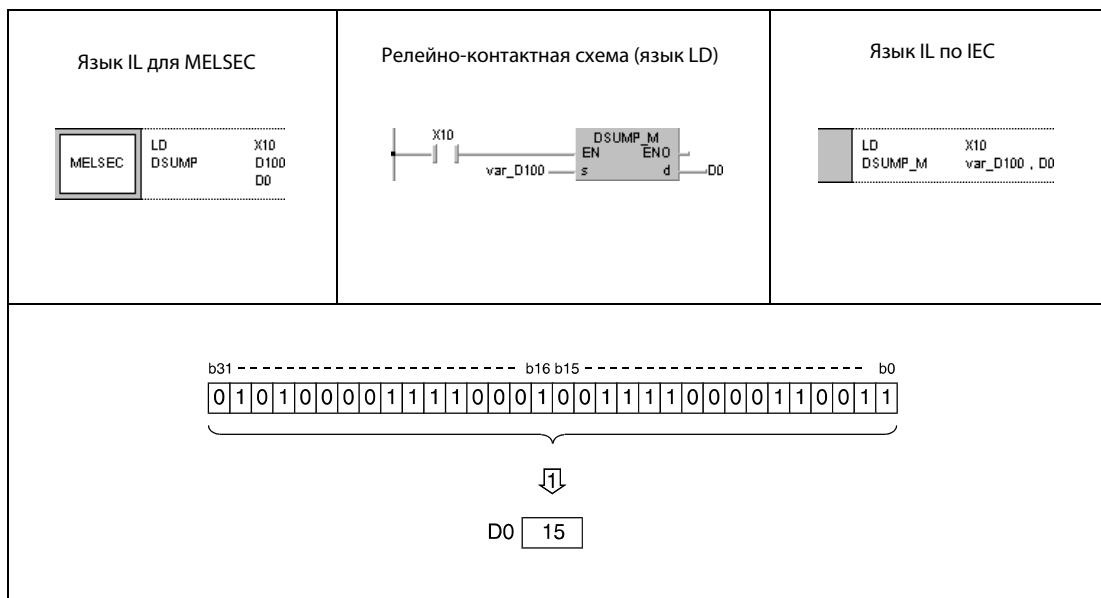
Следующая программа при положительном фронте сигнала X10 определяет сумму включенных (т. е. установленных на 1) входов X8...X17 и записывает результат в D0.



<sup>1</sup> сохранение количества установленных битов в D0

**Пример 2** DSUMP (серии "Q" и "System Q")

Следующая программа при положительном фронте сигнала X10 определяет сумму установленных битов в D100 и D101 и записывает результат в D0.



<sup>1</sup> сохранение количества установленных битов в D0

**ПРИМЕЧАНИЕ** Без объявления переменной в заголовке программного компонента (POU) программа 2 не работоспособна. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

7.5.3 DECO, DECOP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

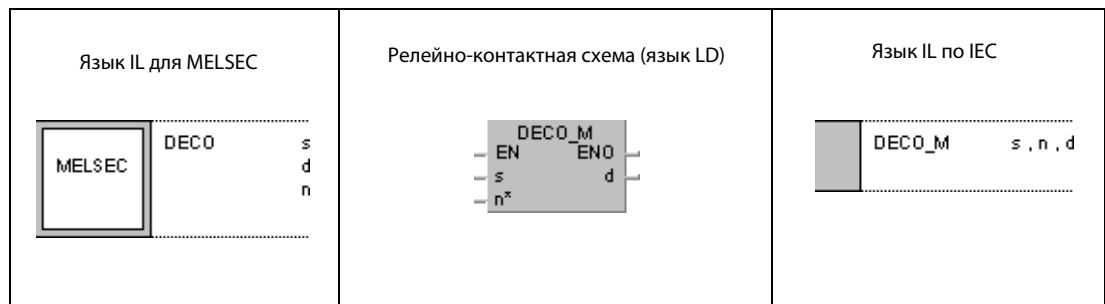
	Операнды															Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011								
	битовые					словные (16 бит)					Константы		Указатели		Уровень													
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V	K	H (16#)	P	I	N		
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●											
d		●	●	●	●	●	●	●	●	●															9	●		●
n																●	●											

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

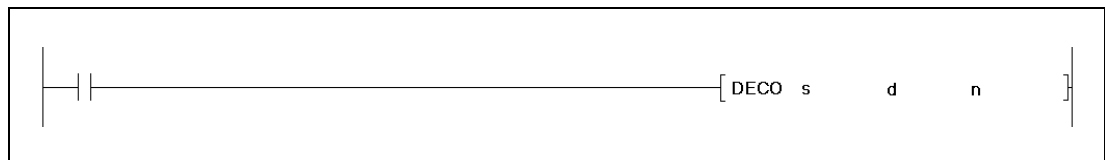
Операнды MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непос. адр. J□□□		Спец. модули U□□□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	●	—	SM0	4
d	●	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
s	Кодированные данные или операнд, в котором сохранены эти данные.	BIN, 16 бит
d	Первый адрес операнда, в котором сохраняется декодированное значение.	адрес
n	Количество битов в кодированных данных	BIN, 16 бит



**Принцип действия**      **Декодировка данных с 8 на 256 бит**  
**DECO    Декodировка данных**

Команда DECO декодирует данные операнда, указанного в s. Содержащееся в нем значение в двоичной кодировке декодируется в десятичное число. Это десятичное число (макс. 256) указывается битом x (bx) операнда, указанного в d. С помощью n устанавливается число адресов в s, содержащих кодированные данные.

В n необходимо указать значение между 1 и 8.

Если n = 0, то команда не обрабатывается и указанный диапазон адресов остается без изменений.

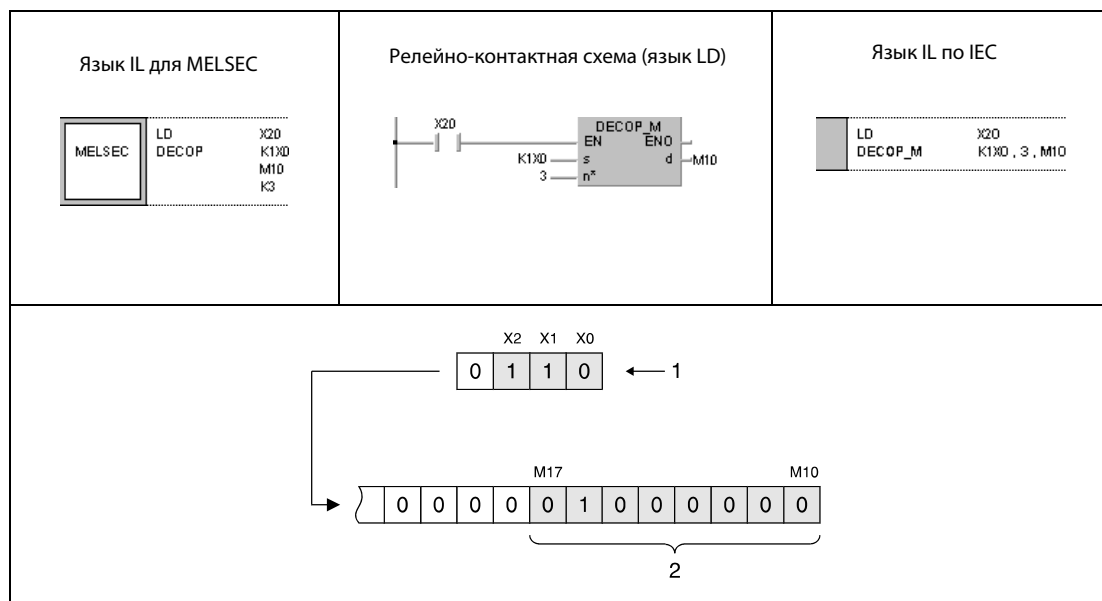
Битовый операнд обрабатывается как отдельный бит, а словный операнд – как 16-битное значение данных.

**Источники ошибок**      В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Значение в n не находится в диапазоне между 1 и 8 (серии "Q"/"System Q" = код ошибки 4100).
- Бит x операнда d находится вне области, предусмотренной для сохранения (серии "Q"/"System Q" = код ошибки 4101).

**Пример**      DECO

Следующая программа при положительном фронте сигнала X20 декодирует данные X0...X2 и сохраняет результат в маркерах M10...M17. Так как данные в X0...X2 содержат двоичное значение 6, в M10...M17 устанавливается бит b6 (M16).



<sup>1</sup> значение 6 в двоичной кодировке  
<sup>2</sup> Если значение в двоичной кодировке указано 4 битами, то для отображения декодированного значения нужны 8 битов.

**ПРИМЕЧАНИЕ**      *Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.*

7.5.4 ENCO, ENCO P

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

	Операнды															Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011							
	битовые					словные (16 бит)					Константы		Указа- тели	Уро- вень													
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V	K	H (16#)	P	I	N	
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●											
d																											●
n																	●	●									

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

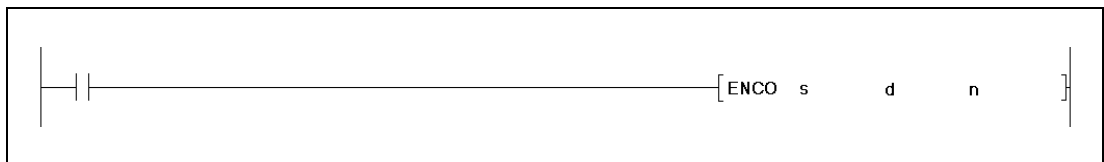
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□□□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	●	●	●	—	—	—	●	—	SM0	4	
d	●	●	●	●	●	●	—	—			
n	●	●	●	●	●	●	●	—			

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
s	Декодированные данные или операнд, в котором сохранены эти данные.	BIN, 16 бит
d	Операнд, в котором сохраняются кодированные данные.	
n	Количество битов, в которых сохраняется кодированное значение.	

**Принцип действия** **Кодировка данных из 256 в 8 бит**

**ENCO Кодировка данных**

Команда ENCO кодирует данные блока данных размером до 256 бит в 8-битный двоичный код. Начальный адрес операндов, данные которого требуется кодировать, устанавливается в s. Установленный в s бит x своей позицией x указывает десятичное значение, которое сохраняется в двоичной кодировке в d. Количество битов операнда, указанного в d и используемого для сохранения закодированного результата, указано в n.

В n необходимо указать значение между 0 и 8.

Если n = 0, то команда не обрабатывается и указанный диапазон адресов остается без изменений.

Битовый операнд обрабатывается как отдельный бит, а словный операнд – как 16-битное значение данных.

Если состояние 1 имеют более одного бита, обработка начинается с самого старшего бита.

**Источники ошибок**

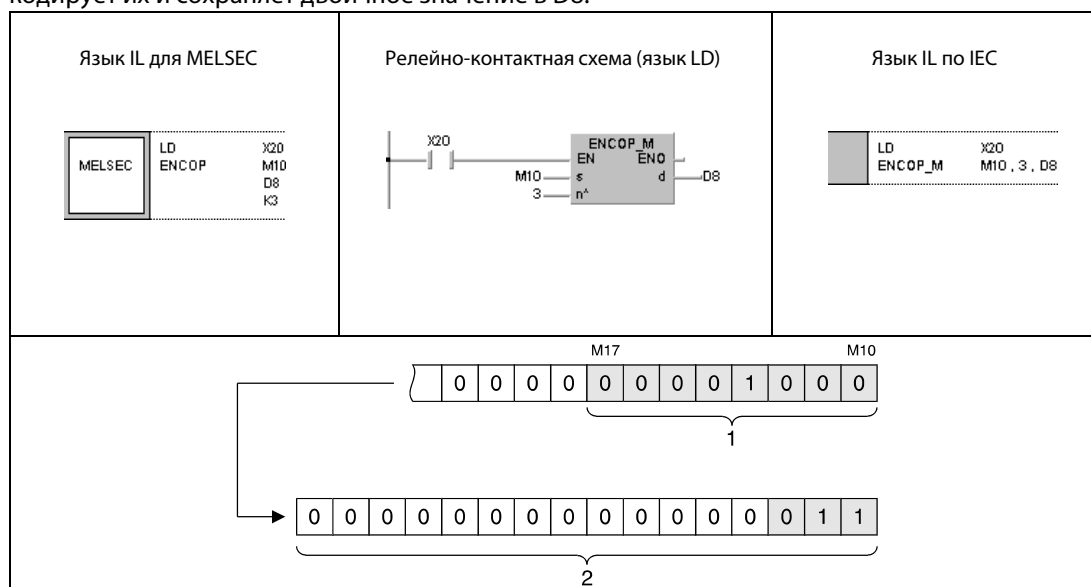
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Значение в n не находится в диапазоне между 0 и 8 ("Q"/"System Q" = код ошибки 4100).
- При выполнении команды ENCO все биты в s до бита x равны 1.
- Значение x, соответствующее установленному биту x в s, находится вне диапазона, который возможно представить в двоичной кодировке с помощью количества битов от 0 до 8 (серии "Q"/"System Q" = код ошибки 4101).
- Все биты в s до бита x совпадают с d (серии "Q"/"System Q" = код ошибки 4100).

**Пример**

ENCO<sup>P</sup>

Следующая программа при положительном фронте сигнала X20 считывает данные из M10...M17, кодирует их и сохраняет двоичное значение в D8.



<sup>1</sup> Если установленный бит кодируется 4 битами двоичной кодировки, может быть отображена область длиной 8 битов.

<sup>2</sup> 3 в двоичной кодировке, что соответствует установленному биту 3 (M13).

**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

7.5.5 SEG, SEGP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
● <sup>1</sup>	● <sup>1</sup>	● <sup>1</sup>	● <sup>1</sup>	●	●

<sup>1</sup> Команда SEG выполняет функцию 7-сегментной декодировки только в том случае, если не установлен специальный маркер M9052. Если специальный маркер M9052 установлен, команда SEG имеет функцию частичного обновления.

Операнды MELSEC A

	Операнды																Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки					
	битовые						словные (16 бит)						Константы		Указа-тели	Уро-вень										
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						K	H (16#)	P	I	N
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					K1	7	●		
d		●	●	●	●	●	●	●	●	●	●	●	●	●	●							● <sup>1</sup>	● <sup>2</sup>			

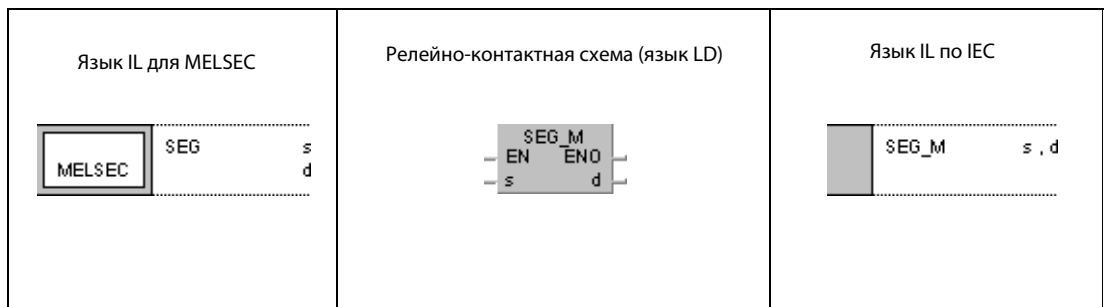
<sup>1</sup> В случае процессоров A3H, A3M или AnN длину блока можно выбирать между K1 и K4. В случае всех прочих процессоров указанная длина блока игнорируется и автоматически обрабатываются блоки длиной K2 (т. е. 8 бит).

<sup>2</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

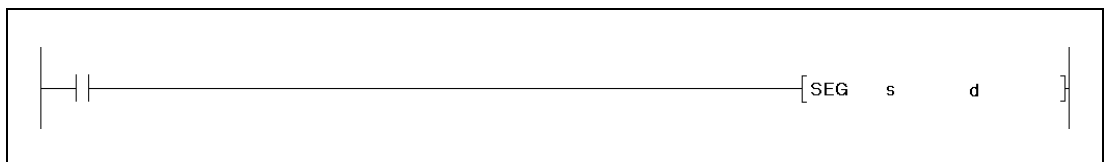
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	●	—	—	3
d	●	●	●	●	●	●	●	—	—	—	

GX IEC Developer



GX Developer



Переменные

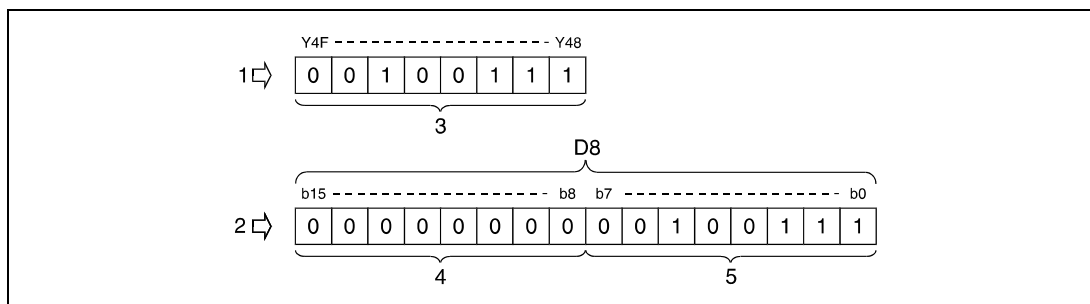
Операнд	Значение	Тип данных
s	Данные, которые требуется преобразовать в 7-сегментные данные, или первый адрес операнда, в котором сохранены эти данные.	BIN, 16 бит
d	Первый адрес операнда, в котором сохраняются 7-сегментные данные.	

**Принцип действия****7-сегментная декодировка****SEG (серии "A", "Q" и "System Q") / SEGP (серии "Q" / "System Q") Преобразование 4-разрядного двоичного значения**

Команда SEG преобразует 4-разрядное двоичное значение в 7-сегментный формат для индикации цифр от 0 до F. Значение данных или начальный адрес преобразуемых данных указывается в s. Семисегментные данные сохраняются в d.

Если 7-сегментные данные выводятся на битовый операнд, то в d следует указать начальный адрес и размер области адресов, в которой должны сохраняться данные. Если d представляет собой словный операнд, то достаточно указать только адрес операнда.

Данные сохраняются в нескольких битовых операндах или в словном операнде по следующей схеме.



<sup>1</sup> битовый операнд

<sup>2</sup> словный операнд

<sup>3</sup> 8 битов

<sup>4</sup> Эти биты всегда имеют значение 0.

<sup>5</sup> 7-сегментные данные

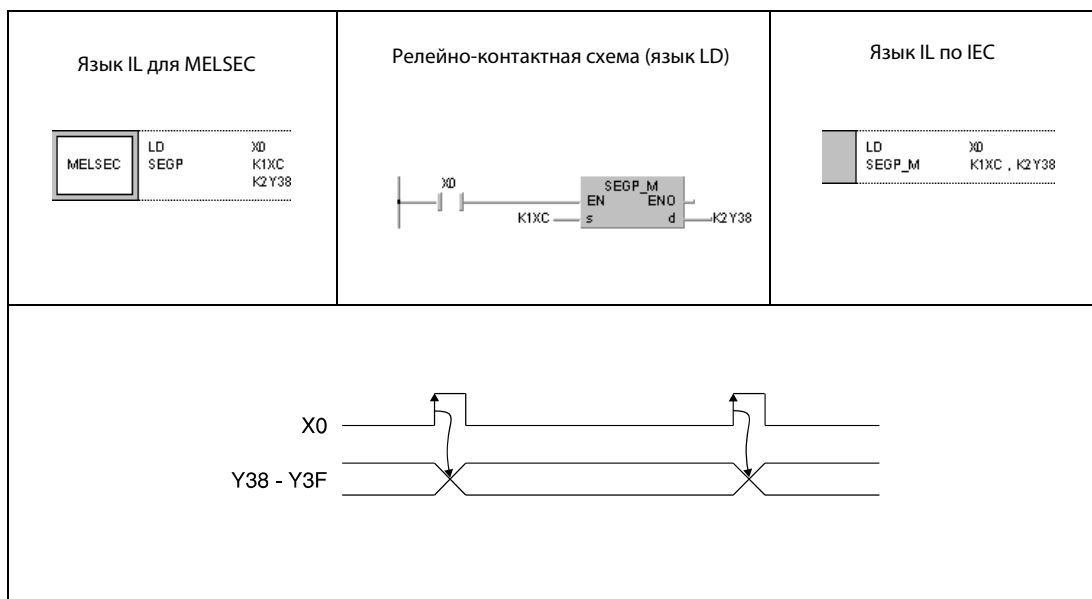
**7-сегментные данные**

В следующей таблице пояснена взаимосвязь между 7-сегментными данными и набором битов исходных данных. Первый бит (b0) 7-сегментных данных отображает либо состояние первого операнда (если данные сохранены в битовом операнде), либо состояние младшего бита словного операнда.

s		Нумерация сегментов	d								Индикация
HEX	Набор битов		B7	B6	B5	B4	B3	B2	B1	B0	
0	0000		0	0	1	1	1	1	1	1	0
1	0001		0	0	0	0	0	1	1	0	1
2	0010		0	1	0	1	1	0	1	1	2
3	0011		0	1	0	0	1	1	1	1	3
4	0100		0	1	1	0	0	1	1	0	4
5	0101		0	1	1	0	1	1	0	1	5
6	0110		0	1	1	1	1	1	0	1	6
7	0111		0	0	1	0	0	1	1	1	7
8	1000		0	1	1	1	1	1	1	1	8
9	1001		0	1	1	0	1	1	1	1	9
A	1010		0	1	1	1	0	1	1	1	A
B	1011		0	1	1	1	1	1	0	0	B
C	1100		0	0	1	1	1	0	0	1	C
D	1101		0	1	0	1	1	1	1	0	D
E	1110		0	1	1	1	1	0	0	1	E
F	1111		0	1	1	1	0	0	0	1	F

**Пример** SEGP (серии "Q" и "System Q")

Следующая программа после включения входа X0 выводит состояния входов XC...XF в 7-сегментном формате на выходы Y38...Y3F. Состояния выходов Y38...Y3F сохраняются до тех пор, пока они не будут перезаписаны новыми данными.



## 7.5.6 DIS, DISP

## Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

## Операнды MELSEC A

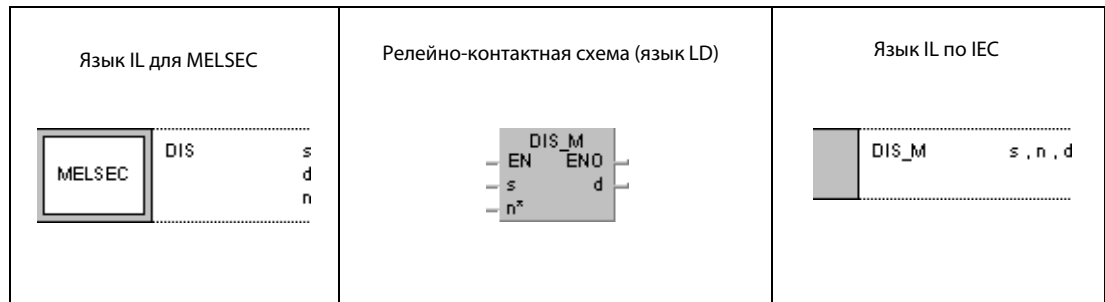
	Операнды															Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011	
	битовые					словные (16 бит)					Константы		Указа-тели	Уро-вень							
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V
s							●	●	●	●	●	●	●	●	●	●	●				
d							●	●	●	●	●										
n																●	●				

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

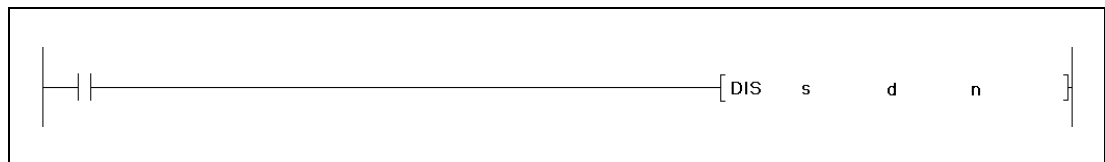
## Операнды MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непос. адр. J□□□		Спец. модули U□□□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	●	—	SM0	4
d	—	●	●	—	—	—	—	—			
n	●	●	●	●	●	●	●	●			

## GX IEC Developer



## GX Developer



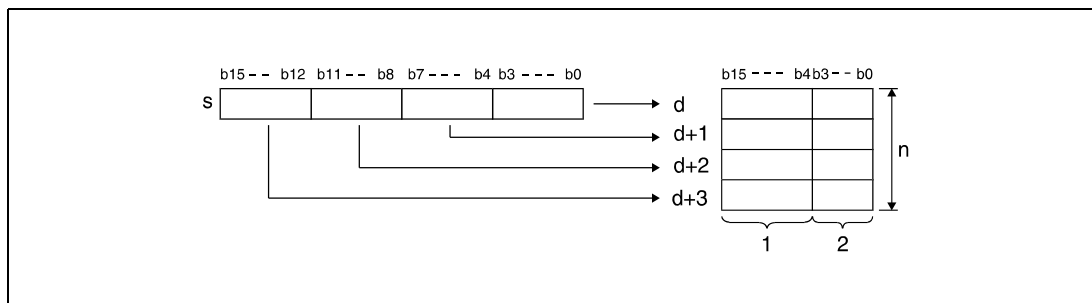
## Переменные

Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранены разделяемые данные.	BIN, 16 бит
d	Первый адрес операнда, в котором сохраняются разделенные данные.	
n	Количество разделяемых 4-битных групп При n = 0 обработка не происходит.	



**Принцип действия****Разделение 16-битных данных****DIS      Разделение 16-битных значений данных**

Команда DIS разделяет 16-битное значение на группы по 4 бита и сохраняет их одну за другой в целевых операндах, количество которых может достигать 4. Разделяемое значение указывается в  $s$ , количество 4-битных групп в  $n$ , а первый целевой адрес в  $d$ . Прочие 4-битные группы записываются в  $d + n$ .



<sup>1</sup> В эти биты записываются нули.

<sup>2</sup> область памяти

Старшие 12 битов целевых операндов, начиная с адреса  $d$ , устанавливаются на 0.

Для  $n$  можно указать значение между 1 и 4 (соответственно, от 4 до 16 битов).

Если  $n = 0$ , то обработка не происходит и указанный адрес операнда остается без изменений.

**Источники ошибок**

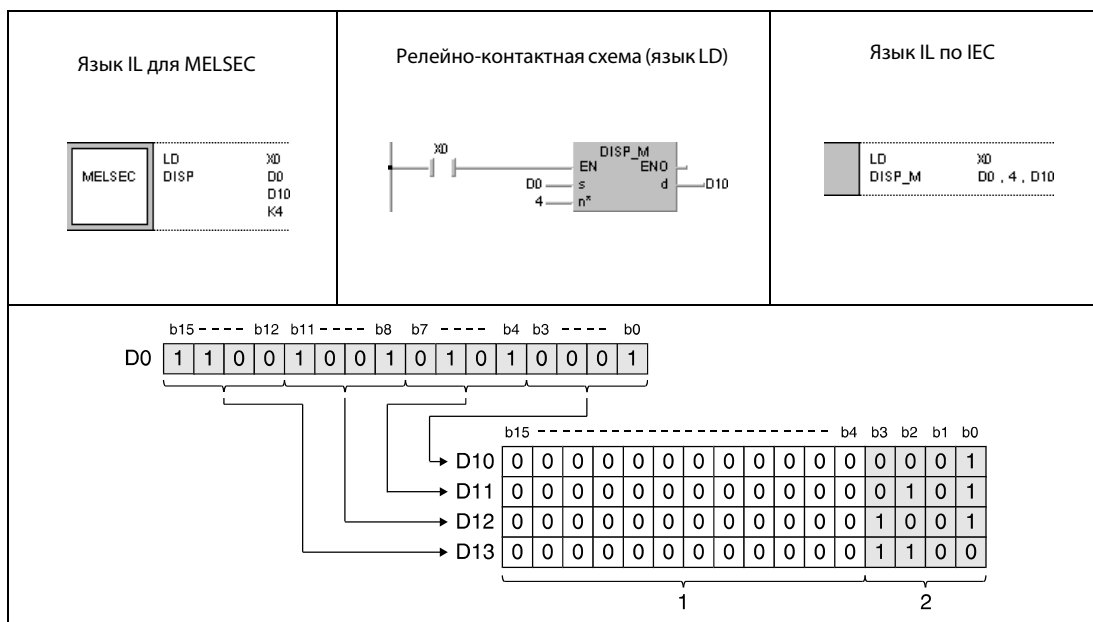
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Значение в  $n$  не находится в диапазоне между 0 и 4 (серии "Q"/"System Q" = код ошибки 4100).
- Область в  $d$ , указанная с помощью  $n$ , превышает предусмотренную для сохранения область операнда (серии "Q"/"System Q" = код ошибки 4101).

## Пример

## DISP

Следующая программа при положительном фронте сигнала X0 разделяет 16-битовое значение в D0 и сохраняет набор битов в виде групп по 4 бита, следующих одна за другой в регистрах с D10 по D13.



<sup>1</sup> В эти биты записываются нули.

<sup>2</sup> область сохранения

### 7.5.7 UNI, UNIP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●☆	●	●

**Операнды MELSEC A**

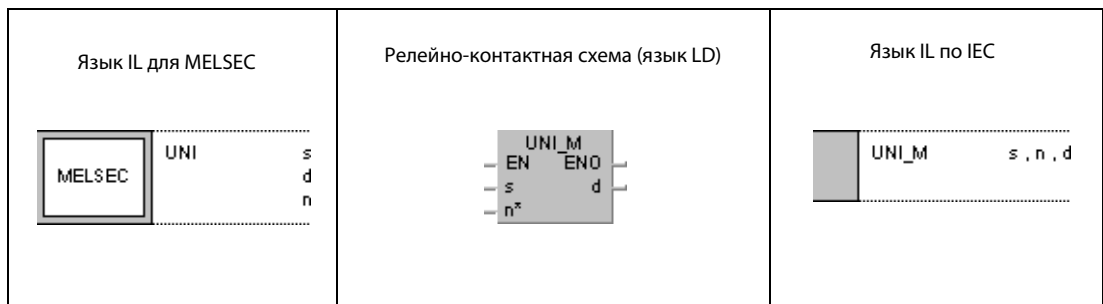
	Операнды															Длина блока K1 ↓ K4	Шагов 9 ↓ 1	Индекс	Флаг пере- носа M9012	Флаг ошибки M9010 M9011	
	битовые					словные (16 бит)						Константы		Указа- тели							Уро- вень
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V
s							●	●	●	●											
d							●	●	●	●	●	●	●	●	●						
n																●	●				

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

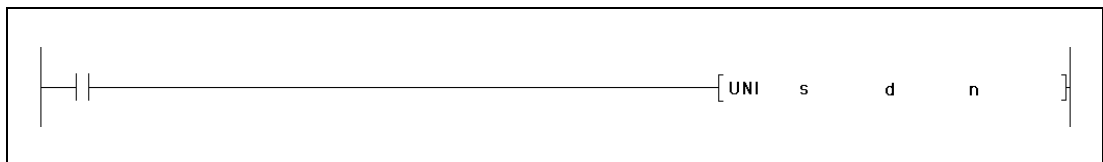
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	●	●	●	—	—	—	—	—	—	SM0	4
d	—	●	●	●	●	●	●	—	—		
n	●	●	●	●	●	●	●	●	—		

**GX IEC Developer**



**GX Developer**

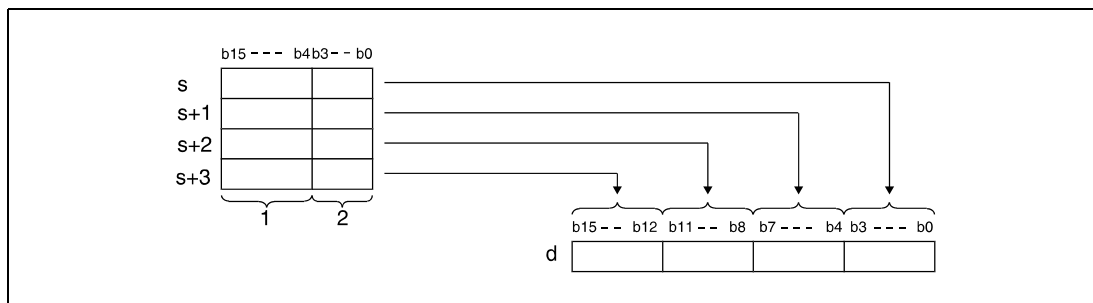


**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранены объединяемые данные.	BIN, 16 бит
d	Первый адрес операнда, в котором сохраняются объединенные данные.	
n	Количество объединяемых 4-битных групп При n = 0 обработка не происходит.	

**Принцип действия****Группирование 16-битных данных****UNI Объединение 16-битных значений данных**

Команда UNI отделяет по 4 младших бита от 16-битных значений, количество которых может достигать четырех, и сохраняет состояния этих битов в одном 16-битном значении. Начальный адрес объединяемых значений указывается в *s*, количество следующих подряд операндов – в *n*, а целевой адрес – в *d*.



<sup>1</sup> Эти биты при обработке не учитываются.

<sup>2</sup> 4-битные группы, сохраняемые в *d*.

Младшие 4 бита начального операнда, начиная с адреса в *d*, устанавливаются на 0.

Для *n* можно указать значение в диапазоне между 1 и 4.

Если *n* = 0, то обработка не происходит и указанный адрес операнда остается без изменений.

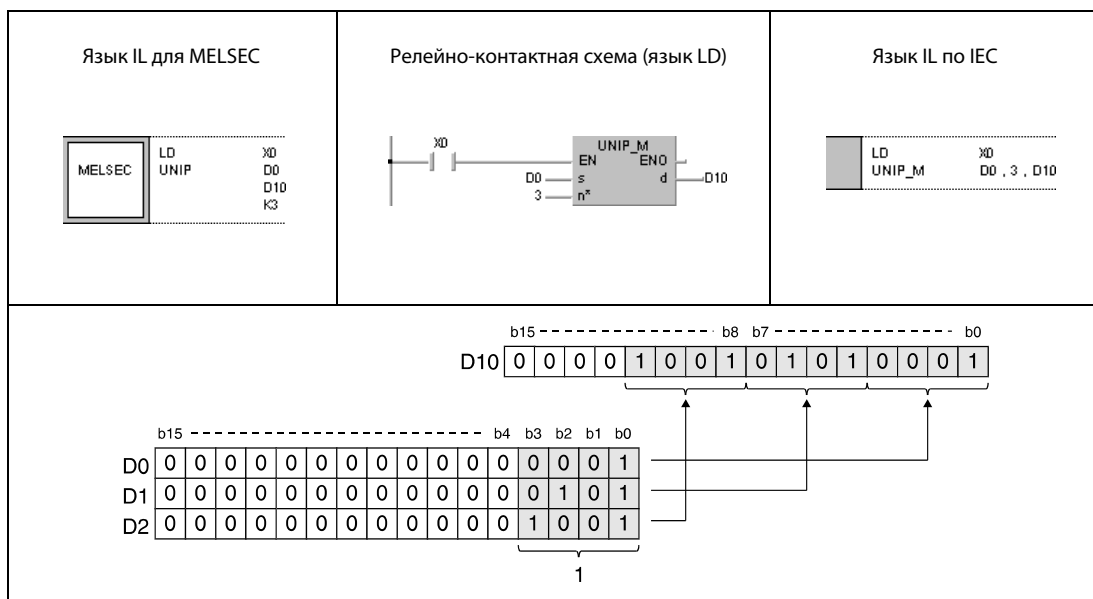
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Значение в *n* не находится в диапазоне между 0 и 4 (серии "Q"/"System Q" = код ошибки 4100).
- Область в *s*, указанная с помощью *n*, превышает предусмотренную для сохранения область операнда (серии "Q"/"System Q" = код ошибки 4101).

**Пример** UNIP

В следующей программе при положительном фронте сигнала X0 первые 4 бита (от b0 до b3) каждого из регистров данных D0...D2 последовательно объединяются в 16-битное значение в D10 (последние 4 разряда заполняются нулями).



<sup>1</sup> сохраняемые в D10 четырехбитные группы

7.5.8 NDIS, NDISP, NUNI, NUNIP

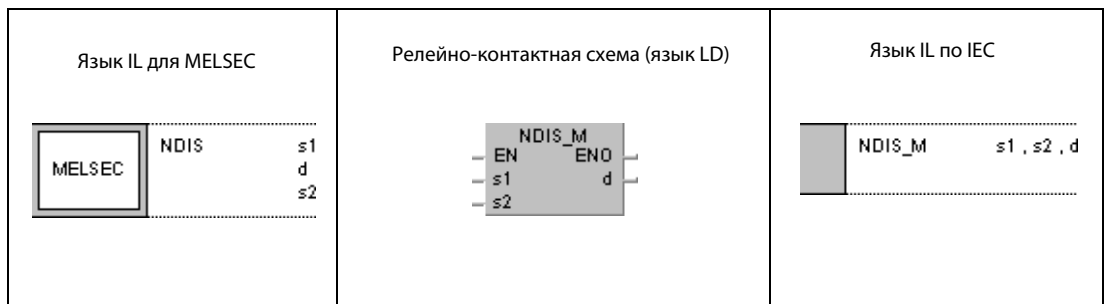
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

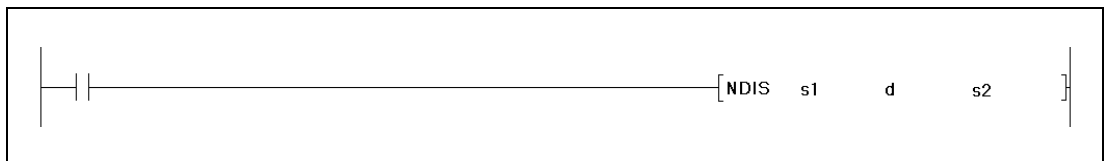
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	SMO	4	
d	—	●	●	—	—	—	—	—			
s2	—	●	●	—	—	—	—	—			

GX IEC Developer



GX Developer



Переменные

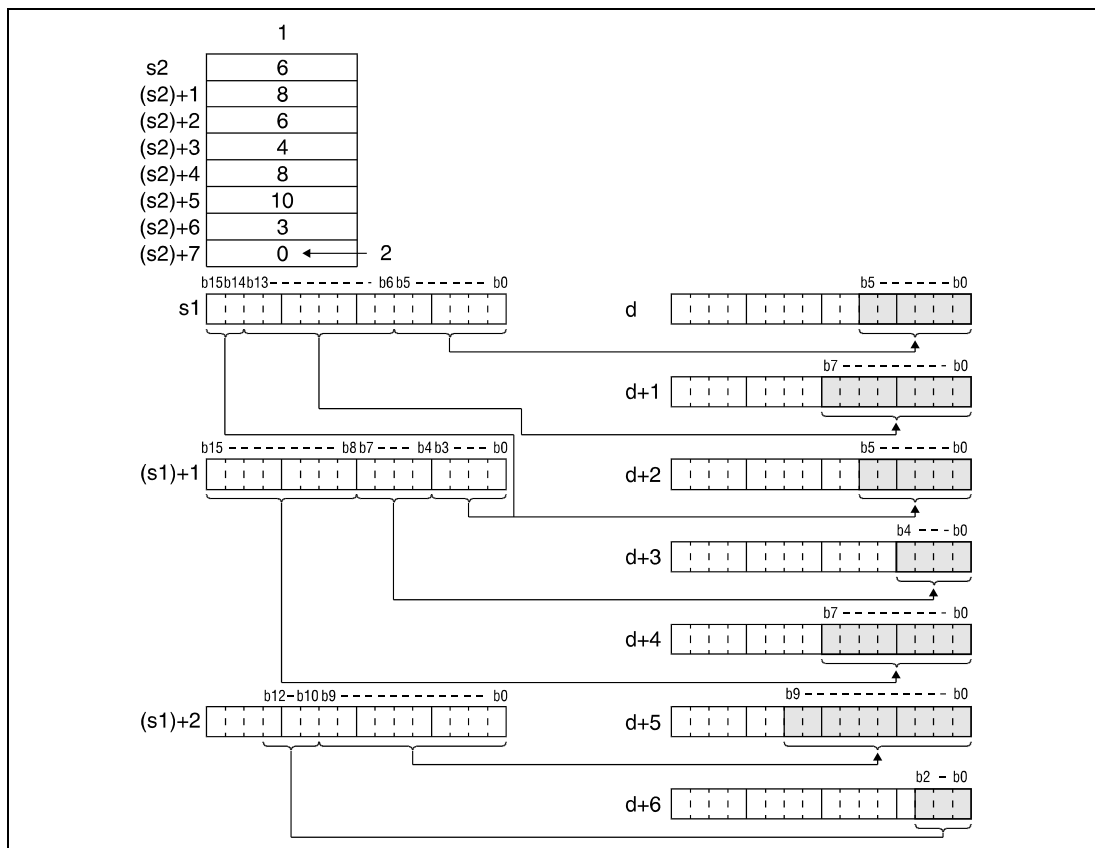
Операнд	Значение	Тип данных
s1	Первый адрес операнда, в котором сохранены разделяемые (объединяемые) данные.	BIN, 16 бит
d	Первый адрес операнда, в котором сохранены разделенные (объединенные) данные.	
s2	Размер разделяемых (объединяемых) групп битов.	

**Принцип действия**

**Разделение и объединение данных в группы битов переменного размера**

**NDIS Разделение данных**

Команда NDIS разделяет данные операндов, начиная с операнда, указанного в s1, на группы битов, размер которых указан в s2. Группы битов сохраняются по отдельности, начиная с операнда, указанного в d.



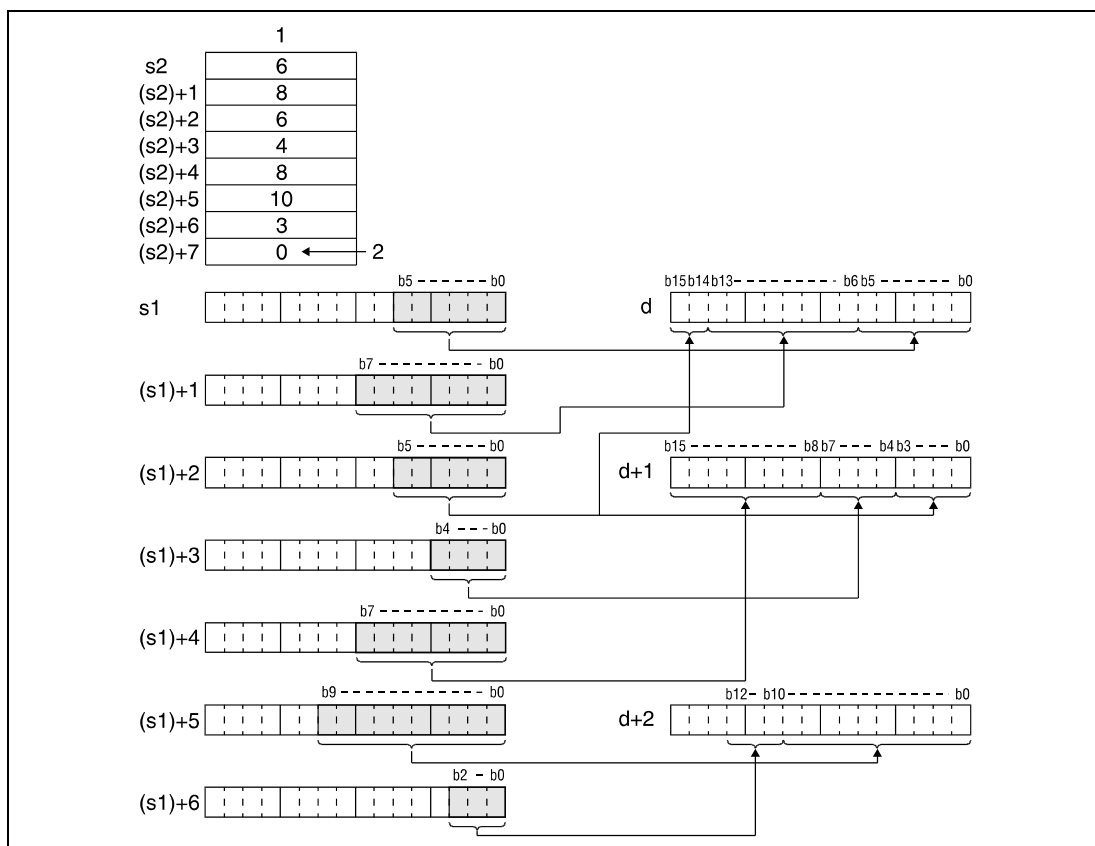
- <sup>1</sup> размер битовых групп
- <sup>2</sup> 0 обозначает конец обработки.

Указанный в s2 размер битовых групп может иметь значение между 1 и 16 битами.

Обрабатываются значения, указанные в s2 и расположенные между первым адресом операнда, указанного в s2, и адресом с записью 0.

**NUNI Объединение данных**

Команда NUNI отделяет битовые группы, размер которых указан начиная с s2, от операндов, хранящихся, начиная с s1, и объединяет эти группы битов в одно значение. Группы битов сохраняются подряд начиная с операнда, указанного в d.



<sup>1</sup> размер битовых групп  
<sup>2</sup> 0 обозначает конец обработки.

Указанный в s2 размер битовых групп может иметь значение между 1 и 16 битами.

Обрабатываются значения, указанные в s2 и расположенные между первым адресом операнда, указанного в s2, и адресом с записью 0.

**Источники ошибок**

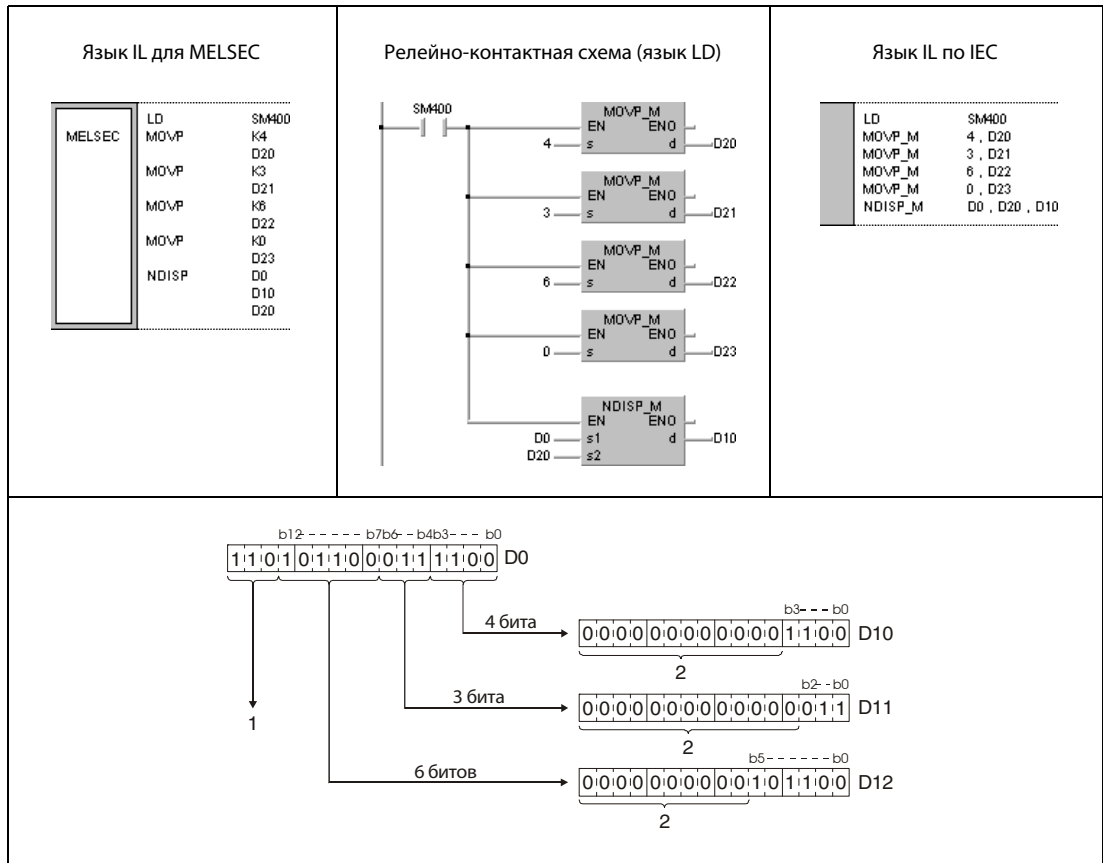
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Группы битов, размер которых указан в s2, в операнде, указанном в s1 или d, находятся вне предусмотренных для сохранения областей операндов (код ошибки 4101).
- Указанный в s2 размер битовых групп не находится в диапазоне между 1 и 16 битами (код ошибки 4100).



**Пример 1** NDISP

Следующая программа при положительном фронте сигнала SM400 отделяет группы битов b0–b3 (4), b4–b6 (3) и b 7–b12 (6) от D0 и сохраняет каждую отдельную группу битов, начиная с группы b0–b3, в регистрах D10...D12. В скобках указан размер групп битов, заданный на адресах с D20 по D22. В регистре D23 должен быть введен ноль (см. "Принцип действия").

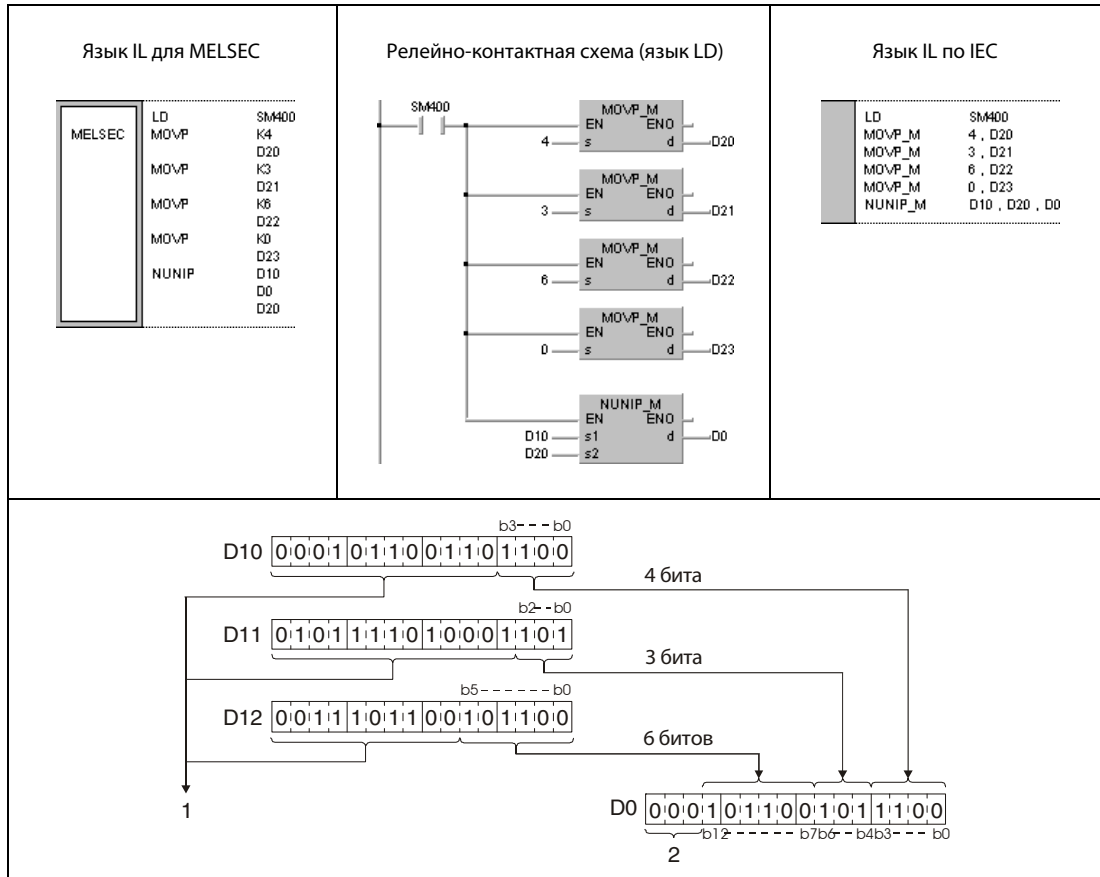


<sup>1</sup> Эти биты при обработке не учитываются.

<sup>2</sup> В эти биты записываются нули.

**Пример 2** NUNIP

Следующая программа при положительном фронте сигнала SM400 отделяет группы битов b0–b3 (4), b0–b2 (3) и b0–b5 (6) от значений регистров D10...D12 и сохраняет эти группы, начиная с группы b0–b3, подряд в регистре D0. В скобках указан размер групп битов, заданный на адресах с D20 по D22. В D23 должен быть введен ноль (см. "Принцип действия").



<sup>1</sup> Эти биты при обработке не учитываются.

<sup>2</sup> В эти биты записываются нули.

### 7.5.9 WTOB, WTOBP, BTOW, BTOWP

**Процессор**

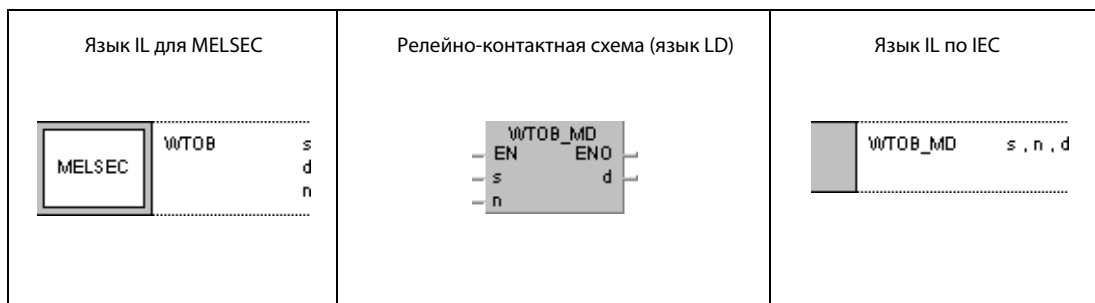
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	●

<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

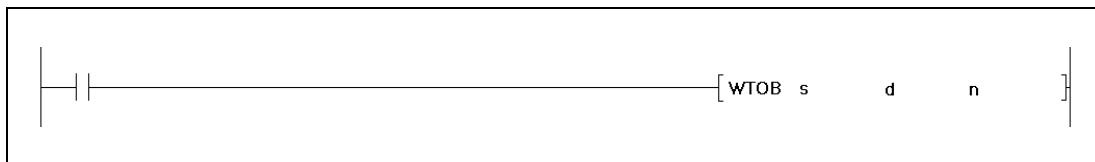
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	—	SMO	4
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

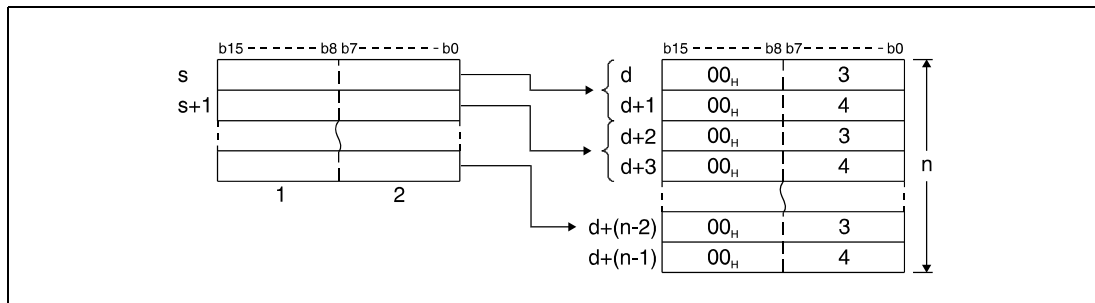
Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранены разделяемые (объединяемые) байты.	BIN, 16 бит
d	Первый адрес операнда, в котором сохранены разделенные (объединенные) байты.	
n	Количество разделяемых (объединяемых) групп байтов.	

**Принцип действия**

**Разделение и объединение данных в группы байтов**

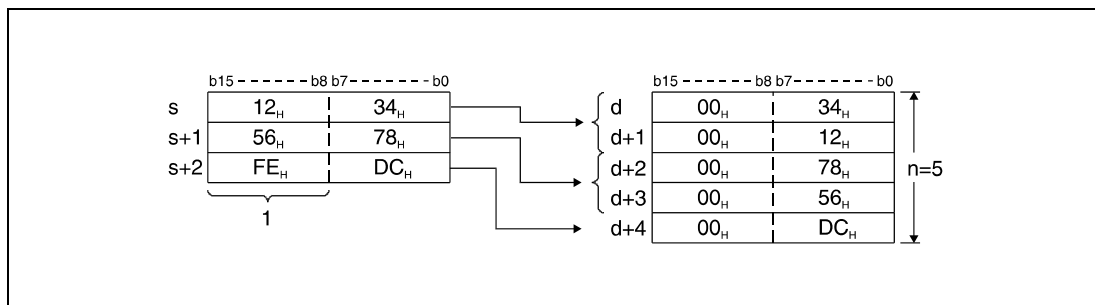
**WTOB Разделение данных**

Команда WTOB разделяет 16-битное значение на группы байтов и сохраняет их одну за другой в целевых операндах. В этой команде разделяемые значения данных указываются в *s*, количество групп байтов в *n*, а первый целевой адрес – в *d*. Прочие группы байтов записываются в *d + n*. Для сохранения используются только младшие байты операндов, указанных в *d*.



- <sup>1</sup> старшие байты
- <sup>2</sup> младшие байты
- <sup>3</sup> данные соответствующих младших байтов
- <sup>4</sup> данные соответствующих старших байтов

Например, если *n = 5*, то от операндов, указанных в данных с *s* по *s + 2*, отделяются по 5 байтов. Эти байты записываются в младшие байты операндов, указанных от *d* до *d + 4*.

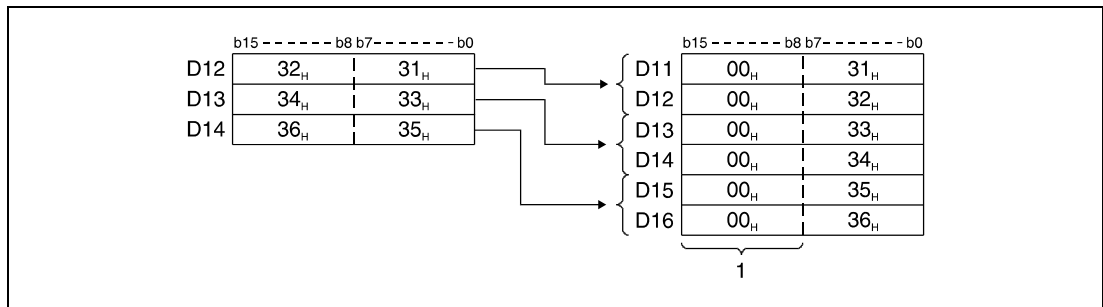


- <sup>1</sup> Эти байты при обработке не учитываются.

Указанное в *n* количество групп байтов автоматически определяет область 16-битных данных в *s* и область памяти для групп байтов в *d*.

Если *n = 0*, то обработка не происходит, и указанные адреса операндов остаются без изменений.

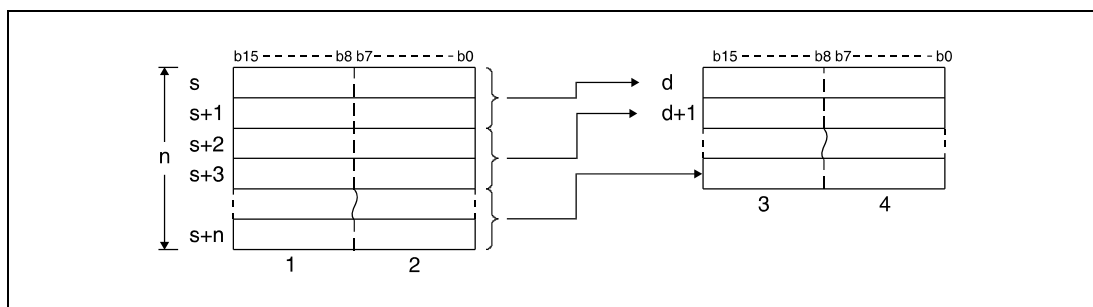
В старшие байты операндов, указанных в d, записывается код "00н".



<sup>1</sup> В эти байты записывается код "00н".

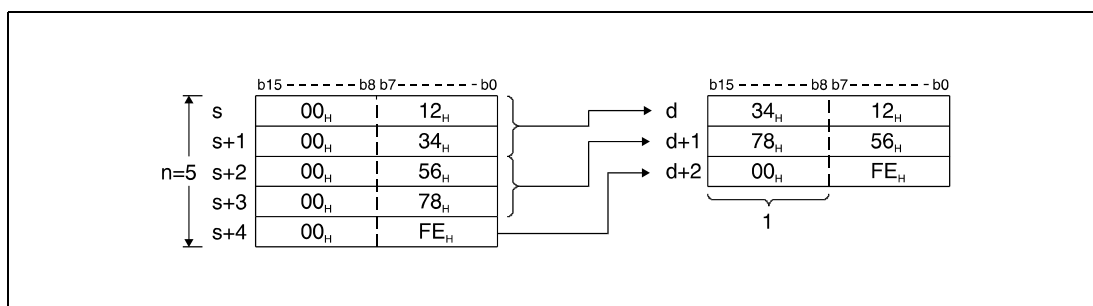
**BTOW Объединение данных**

Команда BTOW отделяет младшие байты от 16-битных значений и сохраняет их состояния вместе в виде 16-битных значений. Начальный адрес объединяемых значений указывается в s, количество групп байтов, следующих подряд – в n, а целевой адрес – в d.



- <sup>1</sup> Эти байты при обработке не учитываются.
- <sup>2</sup> от 1-го до n-ного байта данных
- <sup>3</sup> данные 2-го, 4-го и n-ного байта
- <sup>4</sup> данные 1-го, 3-го и (n – 1)-ного байта

Например, если n = 5, то от операндов, указанных от s до s + 4, отделяются по 5 байтов и эти байты записываются подряд в операнды, указанные с d по d + 2.



- <sup>1</sup> В этот байт записывается значение "00H".

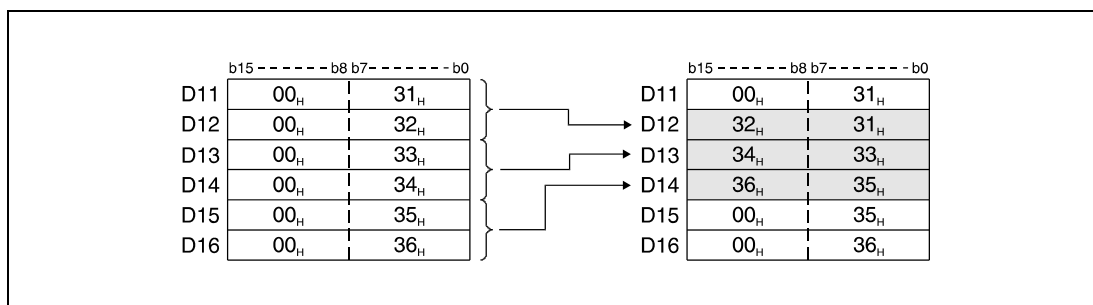
Указанное в n количество групп байтов автоматически определяет область байтовых данных в s и область памяти для байтовых данных в d.

Если n = 0, то обработка не происходит и указанный адрес операнда остается без изменений.

Старшие байты операндов, указанных в s, при обработке не учитываются.

Безошибочная обработка происходит даже в том случае, если область от  $s$  до  $s + n$  перекрывается с областью от  $d$  до  $d + n$ .

На следующей иллюстрации показано сохранение байтов для случая, при котором отдельные младшие байты регистров D11...D16 снова подряд записываются в регистры D12...D14.



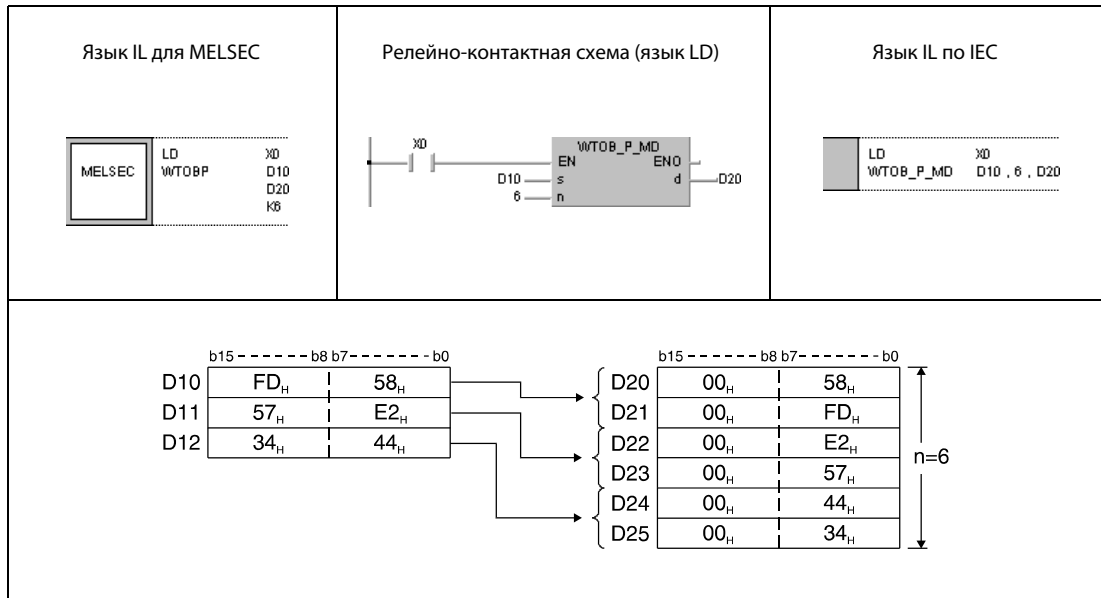
### Источники ошибок

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в  $n$  количество групп байтов, хранящихся начиная с операнда, указанного в  $s$ , находится вне области, предусмотренной для сохранения (код ошибки 4101).
- Указанное в  $n$  количество групп байтов, хранящихся начиная с операнда, указанного в  $d$ , находится вне области, предусмотренной для сохранения (код ошибки 4101).

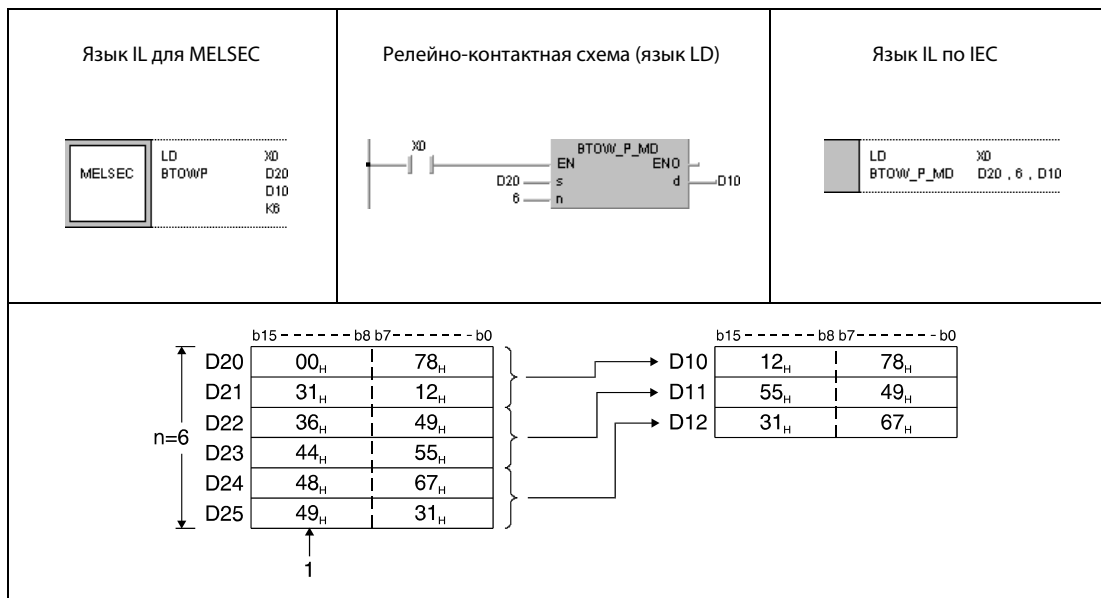
**Пример 1** WTOBP

Следующая программа при положительном фронте сигнала X0 отделяет от данных D10...D12 по 6 байтов и сохраняет эти байты по отдельности в младших байтах регистров с D20 по D25.



**Пример 2** BTOWP

Следующая программа при положительном фронте сигнала X0 отделяет от данных D20...D25 по 6 младших байтов и объединяет их подряд в регистрах с D10 по D12.



<sup>1</sup> Эти байты при обработке не учитываются.



### 7.5.10 MAX, MAXP, DMAX, DMAXP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

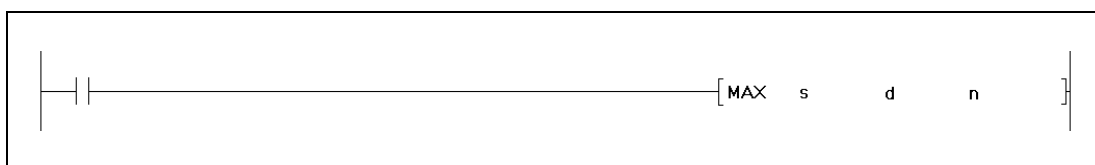
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	SMO	4	
d	—	●	●	—	—	—	—	—			
n	●	●	●	●	●	●	●	—			

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранены проверяемые данные.	BIN, 16/32 бита
d	Первый адрес операнда, в котором сохраняется результат поиска максимального значения.	
n	Количество блоков данных, в которых ведется поиск.	BIN, 16 бит

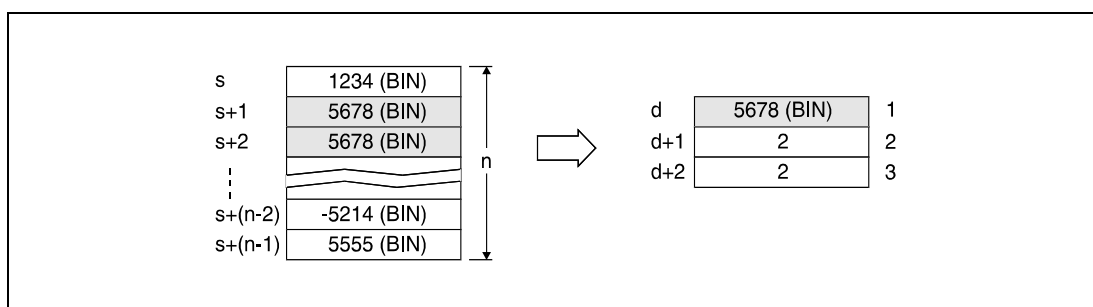
**Принцип действия**

**Поиск максимальных значений в 16/32-битных данных**

**MAX Поиск максимальных значений в 16-битных данных**

Команда MAX ищет максимальное значение в 16-битных блоках данных. Количество блоков данных, в которых ведется поиск, указывается в n. Максимальное значение, найденное в данных от s до s + (n - 1), сохраняется в d.

То место в данных от s до s + (n - 1), в котором впервые встречается максимальное значение, регистрируется, исходя из s = 1, и записывается в d + 1. Количество одинаковых максимальных значений, встречающихся в данных от s до s + (n - 1), записывается в d + 2.

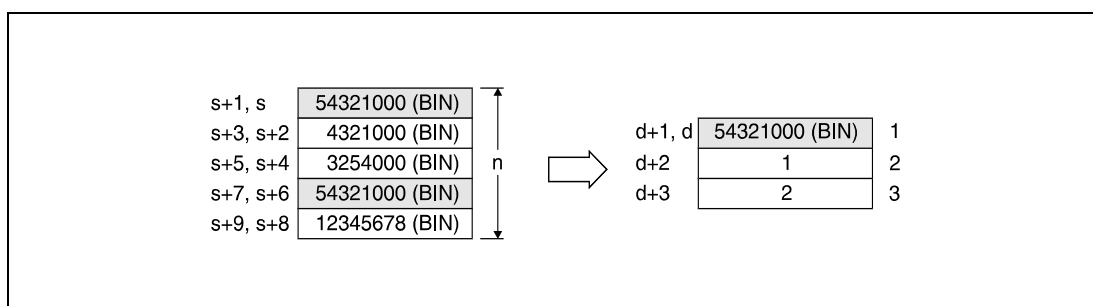


- <sup>1</sup> найденное максимальное значение
- <sup>2</sup> Позиция, в которой это значение встречается впервые.
- <sup>3</sup> количество одинаковых максимальных значений

**DMAX Поиск максимальных значений в 32-битных данных**

Команда DMAX ищет максимальное значение в 32-битных блоках данных. Количество блоков данных, в которых ведется поиск, указывается в n. Максимальное значение, найденное в данных от s до s + (n - 1), сохраняется в d и d + 1.

То место в данных от s до s + (n - 1), в котором максимальное значение встречается впервые, записывается в d + 2. Количество одинаковых максимальных значений, найденных в данных от s до s + (n - 1), записывается в d + 3.



- <sup>1</sup> найденное максимальное значение
- <sup>2</sup> Позиция, в которой это значение встречается впервые.
- <sup>3</sup> количество одинаковых максимальных значений

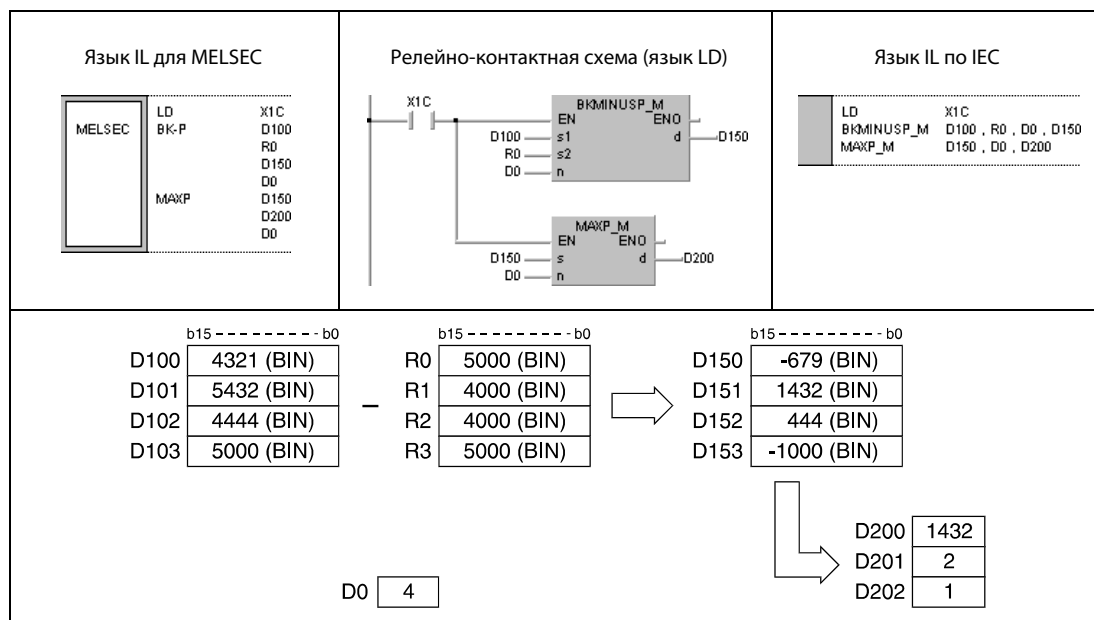
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в n количество блоков данных, сохраненных в операндах, указанных в s, находится вне области, предусмотренной для сохранения (код ошибки 4101).

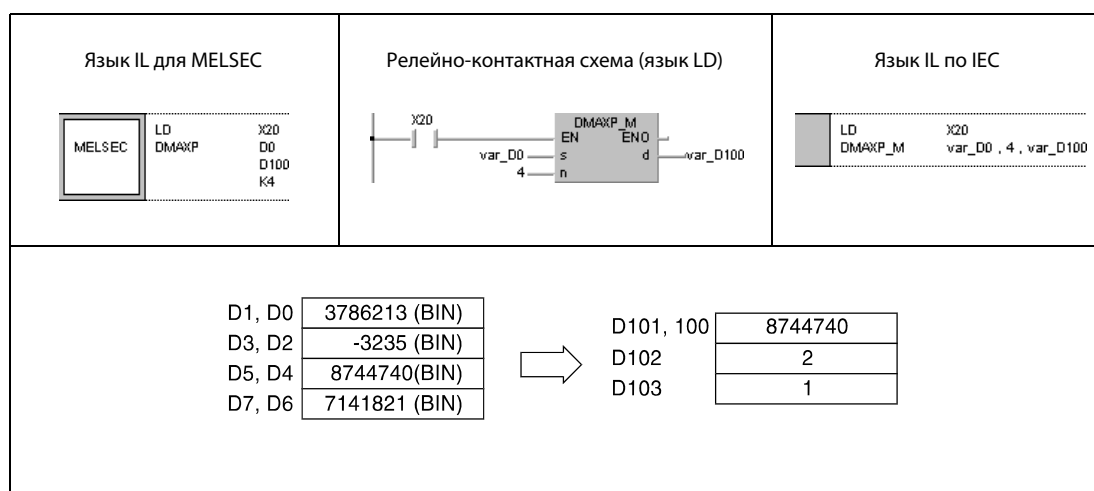
**Пример 1** MAXP

Следующая программа при положительном фронте сигнала X1C вычитает данные в R0...R3 из данных в D100...D103 и сохраняет результат в D150...D153. Количество 16-битных блоков данных (4) указано в D0. В следующем шаге, также при положительном фронте сигнала X1C, ведется поиск максимального значения в регистрах D150...D153. Найденное значение сохраняется в D200, место его первого обнаружения – в D201, а количество одинаковых максимальных значений – в D202.



**Пример 2** DMAXP

Следующая программа при положительном фронте сигнала X20 ищет максимальное значение в 32-битных данных D0...D7 и сохраняет найденное значение в D100 и D101. Место первого обнаружения максимального значения записывается в D102, а количество одинаковых максимальных значений – в D103.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU пример 2 программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

7.5.11 MIN, MINP, DMIN, DMINP

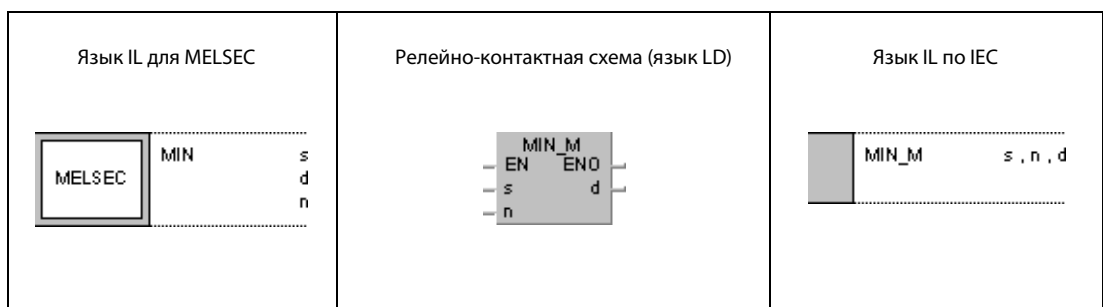
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

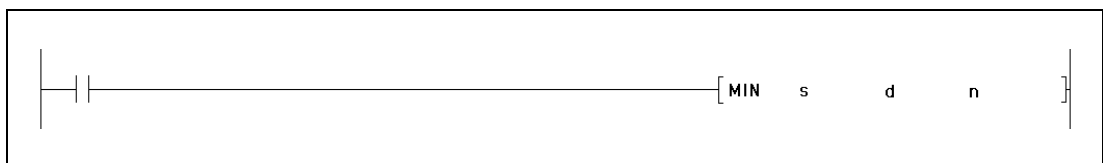
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	SM0	4	
d	—	●	●	—	—	—	—	—			
n	●	●	●	●	●	●	●	●			

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранены проверяемые данные.	BIN, 16/32 бита
d	Первый адрес операнда, в котором сохраняется результат поиска минимального значения.	
n	Количество блоков данных, в которых ведется поиск.	BIN, 16 бит

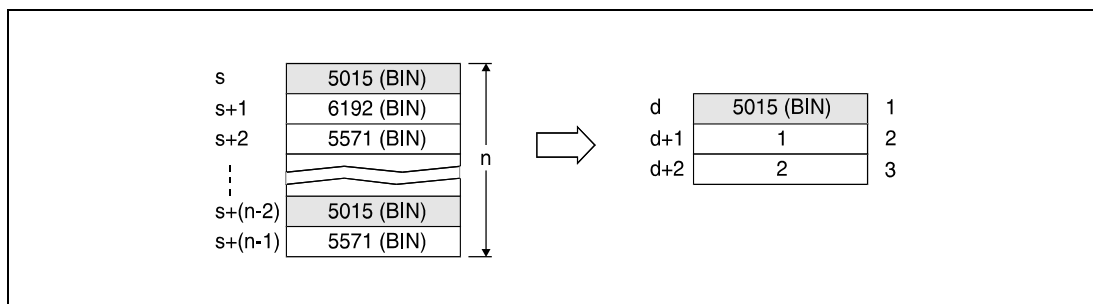
**Принцип действия**

**Поиск минимальных значений в 16/32-битных данных**

**MIN Поиск минимальных значений в 16-битных данных**

Команда MIN ищет минимальное значение в 16-битных блоках данных. Количество блоков данных, в которых ведется поиск, указывается в n. Минимальное значение, найденное в данных от s до s + (n - 1), сохраняется в d.

Место в данных от s до s + (n - 1), в котором минимальное значение встречается впервые, регистрируется, исходя из s = 1, и записывается в d + 1. Количество одинаковых минимальных значений, имеющих в данных от s до s + (n - 1), записывается в d + 2.

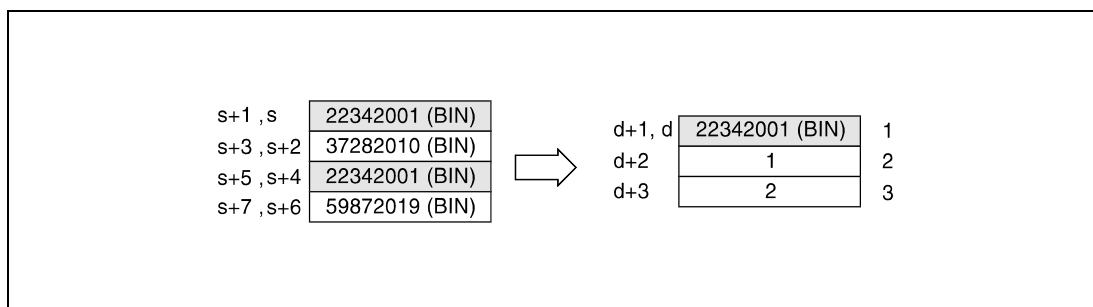


- <sup>1</sup> найденное минимальное значение
- <sup>2</sup> Позиция, в которой это значение встречается впервые.
- <sup>3</sup> количество одинаковых минимальных значений

**DMIN Поиск минимальных значений в 32-битных данных**

Команда DMIN ищет минимальное значение в 32-битных блоках данных. Количество блоков данных, в которых ведется поиск, указывается в n. Минимальное значение, найденное в данных от s до s + (n - 1), сохраняется в d и d + 1.

То место в данных от s до s + (n - 1), в котором минимальное значение встречается впервые, записывается в d + 2. Количество одинаковых минимальных значений, имеющих в данных от s до s + (n - 1), записывается в d + 3.



- <sup>1</sup> найденное минимальное значение
- <sup>2</sup> Позиция, в которой это значение встречается впервые.
- <sup>3</sup> количество одинаковых минимальных значений

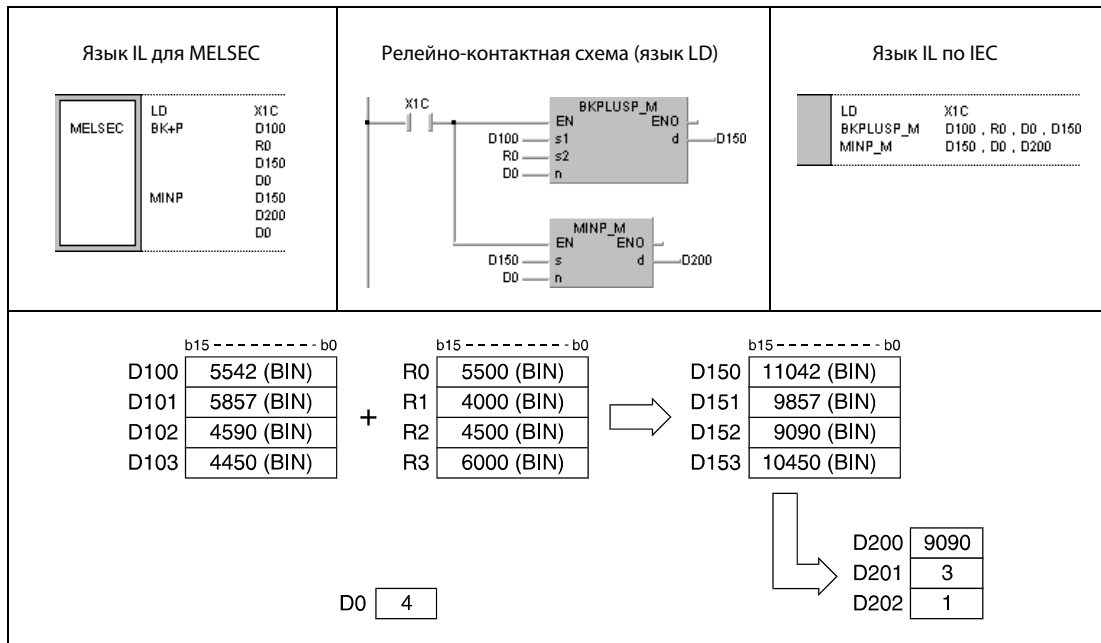
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в n количество 32-битных блоков данных, хранящихся начиная с операндов, указанных в s и s + 1, находится вне области, предусмотренной для сохранения (код ошибки 4101).

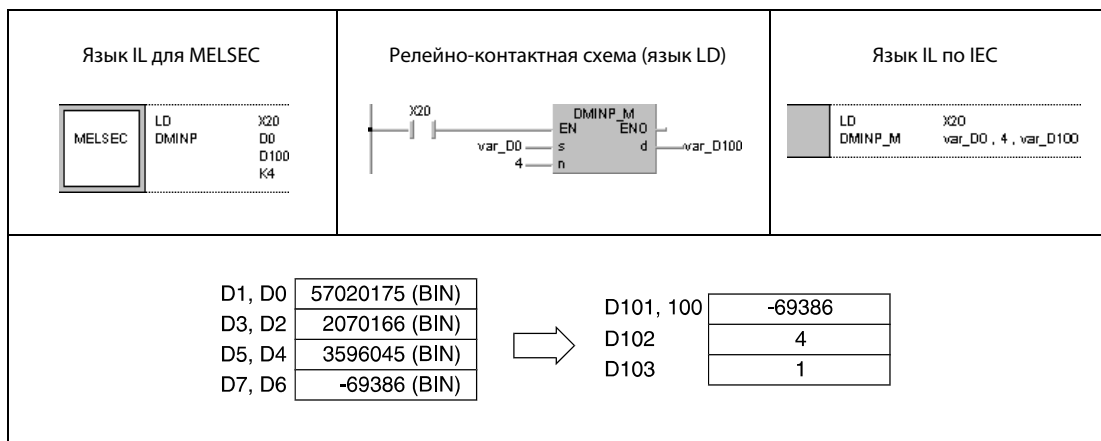
**Пример 1** MINP

Следующая программа при положительном фронте сигнала X1C суммирует данные D100...D103 с данными R0...R3 и сохраняет результат в регистрах D150...D153. Количество 16-битных блоков данных (4) указано в D0. В следующем шаге, также при положительном фронте сигнала X1C, в регистрах D150...D153 ведется поиск минимального значения. Найденное значение сохраняется в D200, место его первого обнаружения в D201, а количество одинаковых минимальных значений – в D202.



**Пример 2** DMINP

Следующая программа при положительном фронте сигнала X20 ищет минимальное значение в 32-битных данных в D0...D7 и сохраняет найденное значение в D100 и D101. Место его первого обнаружения сохраняется в D102, а количество одинаковых минимальных значений – в D103.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU пример 2 программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

**7.5.12 SORT, SORTP, DSORT, DSORTP**

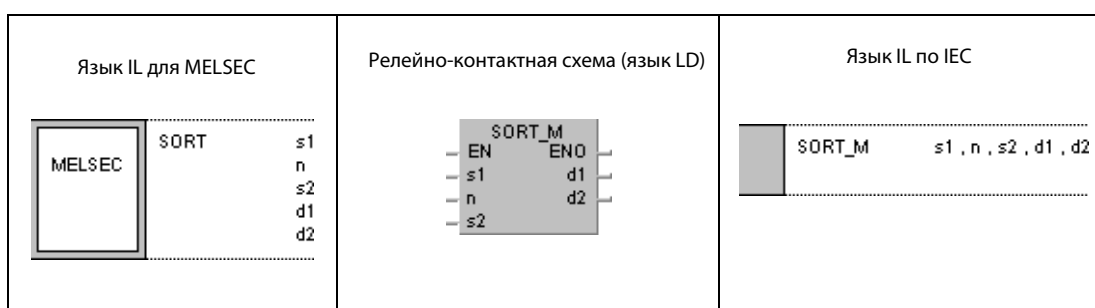
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

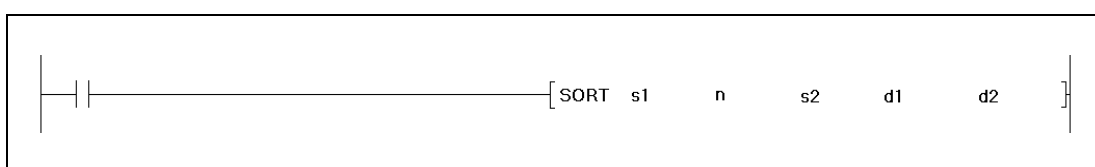
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	SM0	6	
n	●	●	●	—	—	—	—	—			
s2	●	●	●	●	●	●	●	—			
d1	●	—	—	●	●	●	●	—			
d2	—	●	●	—	—	—	—	—			

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s1	Первый адрес операнда, в котором сохранены сортируемые данные.	BIN, 16/32 бита
n	Количество сортируемых блоков данных.	BIN, 16 бит
s2	Количество блоков данных, сравниваемых во время сортировки.	BIN, 16 бит
d1	Адрес бита, устанавливаемого по окончании сортировки.	бит
d2	Операнд для внутрисистемного использования.	BIN, 16 бит

**Принцип действия**

**Сортировка 16/32-битных данных**

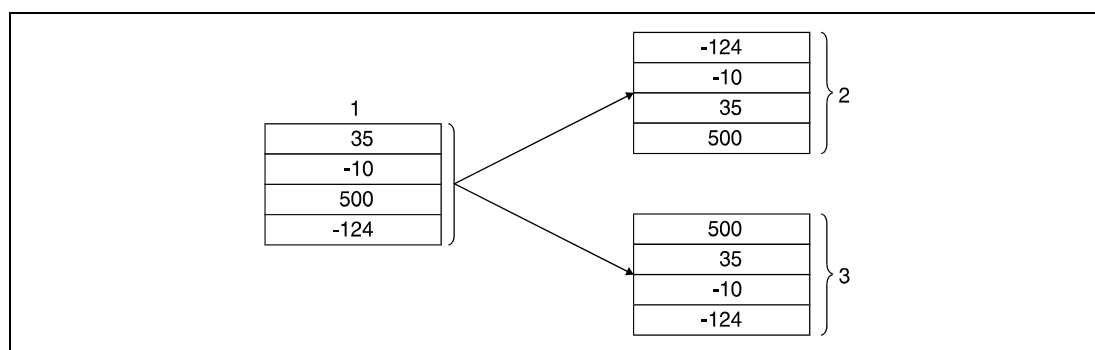
**SORT Сортировка 16-битных данных**

Команда SORT сортирует указанное в n количество 16-битных данных, указанных в s1, в возрастающей или убывающей последовательности.

Направление сортировки устанавливается с помощью маркера диагностики SM703.

SM703 выключен: возрастающее направление сортировки

SM703 включен: убывающее направление сортировки



<sup>1</sup> сортируемые данные

<sup>2</sup> данные, отсортированные по возрастанию (SM703 = выключен)

<sup>3</sup> данные, отсортированные по убыванию (SM703 = включен)

Для выполнения команды SORT необходимы несколько проходов. Количество требуемых проходов определяется путем деления максимального количества проходов на указанное в s2 количество 16-битных данных, сравниваемых за один проход, (дробь округляется вверх). Увеличение указанного в s2 количества 16-битных данных снижает число требуемых проходов, однако увеличивает время обработки.

Требуемое количество проходов до завершения операции сортировки рассчитывается по следующему уравнению:

$$\text{Требуемое количество проходов сортировки} = ((n) \times (n - 1)) / (2 \times (s2))$$

Например, если в это уравнение подставить данные n = 10 и s2 = 1, то для завершения операции сортировки нужны 45 проходов.

В случае n = 10 и s2 = 2 результат равен 22.5 проходам. После округления вверх окончательный результат гласит 23 прохода.

Указанный в d1 бит во время операции сортировки сброшен и устанавливается лишь после ее завершения. Этот бит остается установленным и его необходимо сбрасывать с помощью отдельной команды.

Указанные в d2 + 0 и d2 + 1 операнды используются для внутрисистемной обработки во время сортировки. Поэтому эти операнды не следует изменять путем программирования.

Если во время сортировки изменено значение в n, обработка происходит на основе количества 16-битных данных, установленного на данный момент.

При сбросе условия выполнения обработка прерывается. При повторной установке условия выполнения операция сортировки начинается заново.



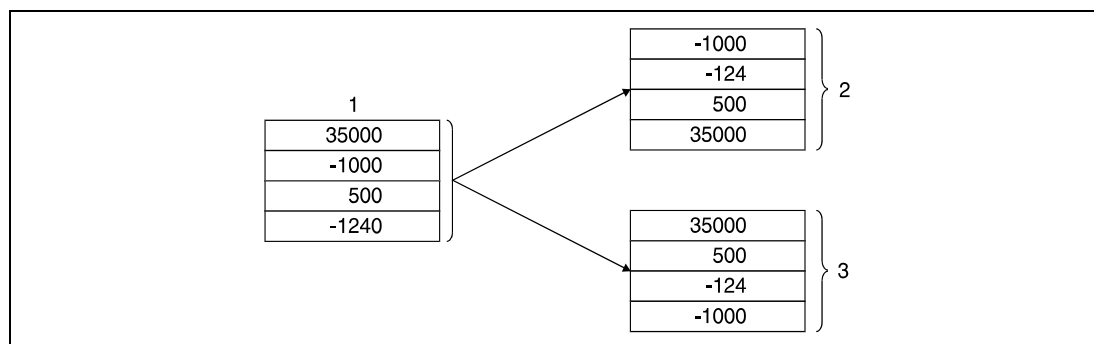
**DSORT Сортировка 32-битных данных**

Команда SORT сортирует указанное в  $n$  количество 32-битных данных, указанных в  $s1$ , в возрастающей или убывающей последовательности.

Направление сортировки устанавливается с помощью маркера диагностики SM703.

SM703 выключен: возрастающее направление сортировки

SM703 включен: убывающее направление сортировки



<sup>1</sup> сортируемые данные

<sup>2</sup> данные, отсортированные по возрастанию (SM703 = выключен)

<sup>3</sup> данные, отсортированные по убыванию (SM703 = включен)

Для выполнения команды DSORT необходимы несколько проходов. Количество требуемых проходов сортировки определяется путем деления максимального количества проходов на указанное в  $s2$  количество 32-битных данных, сравниваемых за один проход, (дроби округляются вверх). Увеличение указанного в  $s2$  количества 32-битных данных уменьшает число требуемых проходов сортировки, однако увеличивает время обработки.

Требуемое количество проходов до завершения операции сортировки рассчитывается по следующему уравнению:

$$\text{Требуемое количество проходов сортировки} = ((n) \times (n - 1)) / (2 \times (s2))$$

Например, если в это уравнение подставить данные  $n = 10$  и  $s2 = 1$ , то для завершения операции сортировки нужны 45 проходов.

В случае  $n = 10$  и  $s2 = 2$  результат равен 22.5 проходам. После округления вверх окончательный результат гласит 23 прохода.

Указанный в  $d1$  бит во время операции сортировки сброшен и устанавливается лишь после ее завершения. Этот бит остается установленным и его необходимо сбрасывать с помощью отдельной команды.

Указанные в  $(d2) + 0$  и  $(d2) + 1$  операнды используются для внутрисистемной обработки во время сортировки. Поэтому эти операнды не следует изменять путем программирования.

Если во время сортировки изменено значение в  $n$ , то обработка происходит на основе количества 32-битных данных, установленного на данный момент.

При сбросе условия выполнения обработка прерывается. При повторной установке условия выполнения операция сортировки начинается заново.

**Источники ошибок**

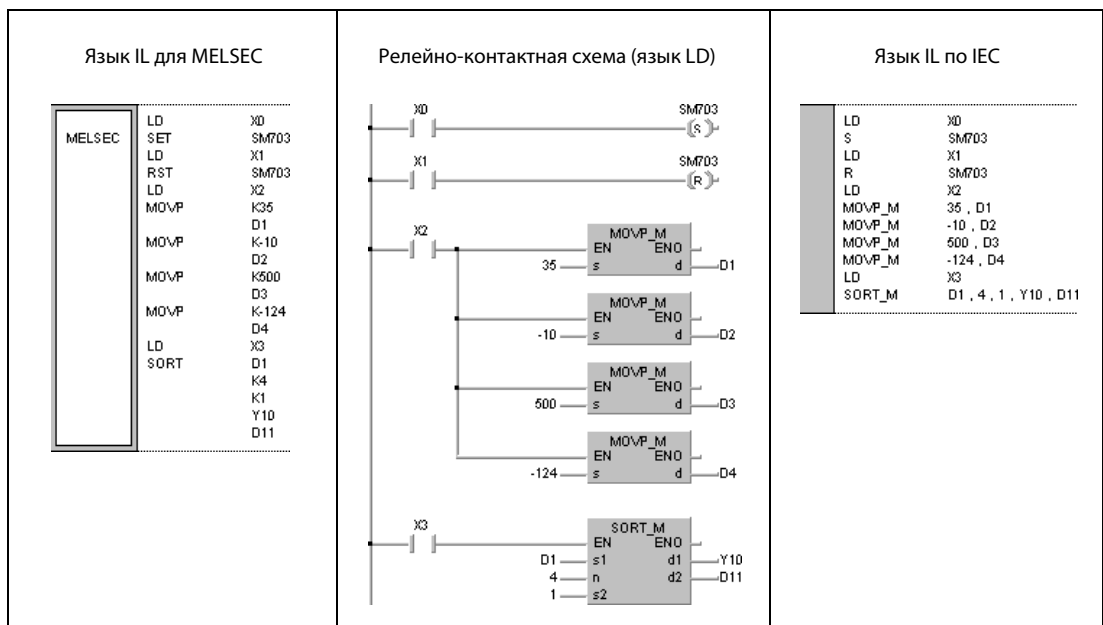
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанная с помощью n или 2 x n область операнда, указанного в s1, находится вне области, предусмотренной для сохранения (код ошибки 4101).
- В s2 указан 0 или отрицательное значение (код ошибки 4100).
- Значение в d2 + 0 больше значения в n (код ошибки 4101).
- Значение в d2 + 1 больше значения в d2+0 (код ошибки 4101).

**Пример**

**SORT**

Следующая программа при включенном входе X3 сортирует 16-битные данные в регистрах D1...D4. В первом шаге при положительном фронте сигнала X2 в регистры D1...D4 загружаются значения 35, -10, 500 и -124. Затем происходит сортировка. Направление сортировки задается через входы X0 (установка маркера SM703) и X1 (сброс маркера SM703). По окончании сортировки устанавливается выход Y10.



### 7.5.13 WSUM, WSUMP

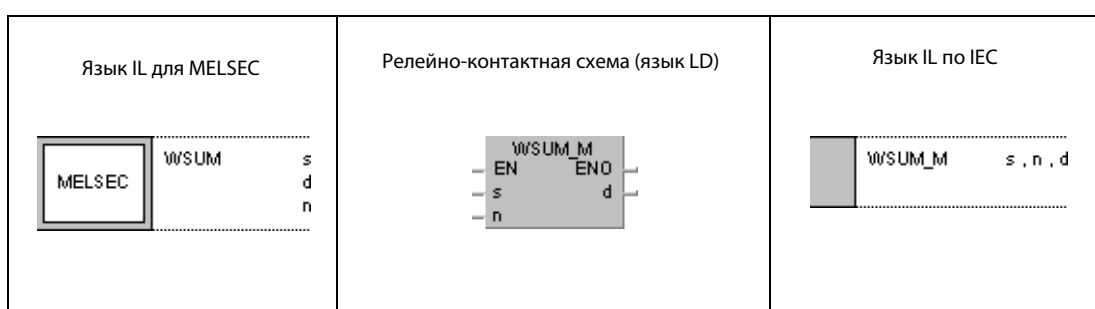
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

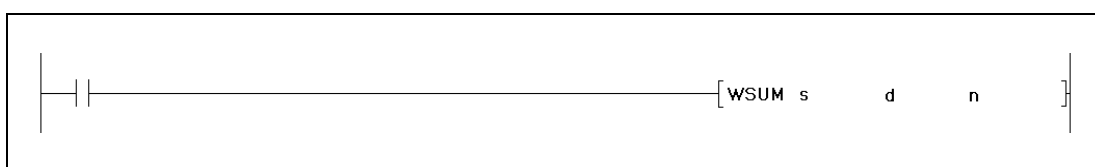
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\V□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	SM0	4	
d	●	●	●	●	●	●	●	—			
n	●	●	●	●	●	●	●	—			

**GX IEC Developer**



**GX Developer**



**Переменные**

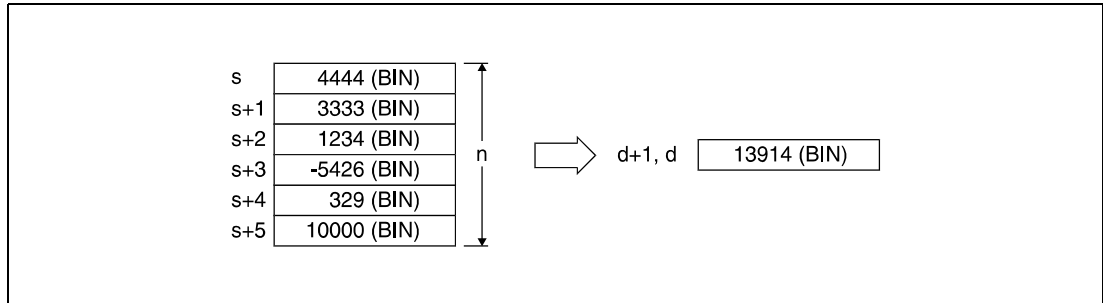
Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранены суммируемые данные.	BIN, 16 бит
d	Первый адрес операнда, в котором сохраняется результат.	BIN, 32 бита
n	Количество суммируемых блоков данных.	BIN, 16 бит

**Принцип действия**

**Вычисление суммы 16-битных двоичных данных**

**WSUM Команда суммирования**

Команда WSUM суммирует указанное количество (n) блоков 16-битных двоичных данных в операнде, указанном в s. Результат сохраняется в операнде, указанном в d.



**Источники ошибок**

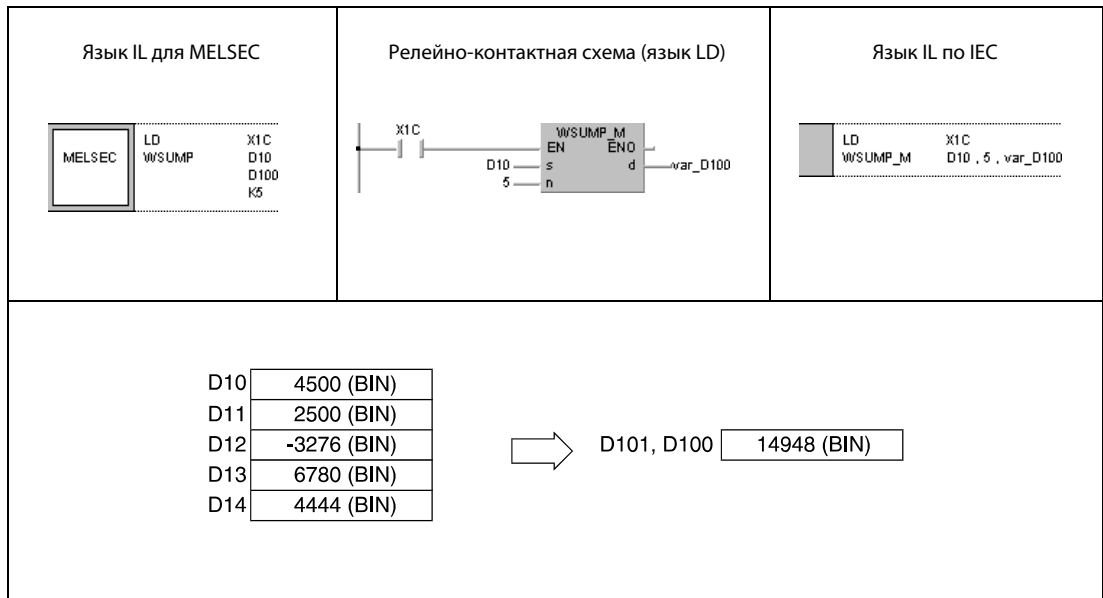
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанная с помощью n область в операнде, указанном в s, находится вне области, предусмотренной для сохранения (код ошибки 4101).

**Пример**

**WSUMP**

Следующая программа при положительном фронте сигнала X1C суммирует 16-битные блоки двоичных данных в регистрах D10...D14 и сохраняет результат в регистрах D100 и D101.



### 7.5.14 DWSUM, DWSUMP

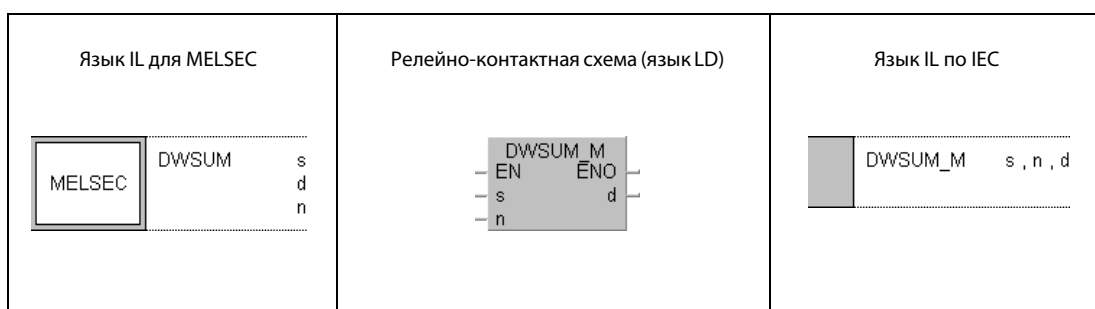
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

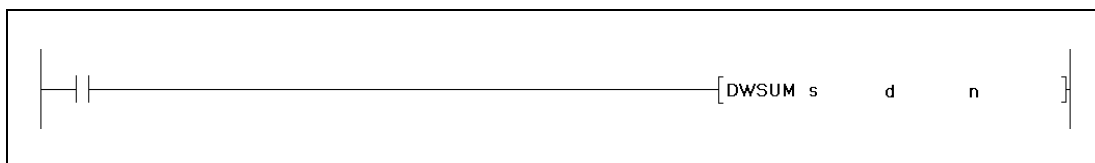
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	—	SMO	4
d	●	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

**GX IEC Developer**



**GX Developer**

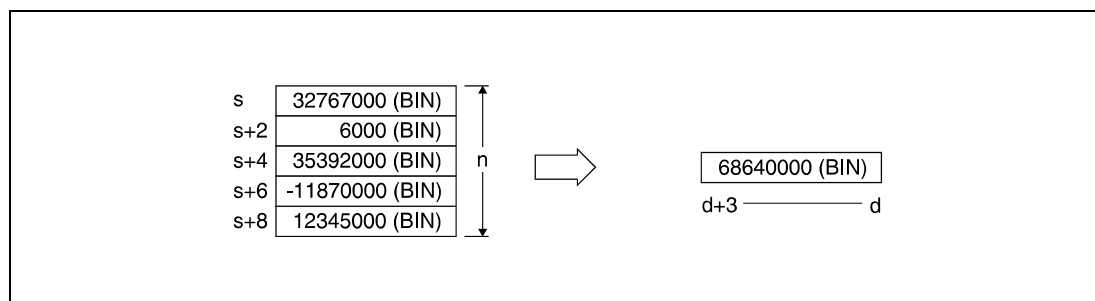


**Переменные**

Операнд	Значение	Тип данных	
		MELSEC	IEC
s	Первый адрес операнда, в котором сохранены суммируемые данные.	BIN, 32 бита	ANY32
d	Первый адрес операнда, в котором сохраняется результат.	BIN, 64 бита	массив [1..4] данных типа ANY16
n	Количество суммируемых блоков данных.	BIN, 16 бит	ANY16

**Принцип действия****Вычисление суммы 32-битных двоичных данных****DWSUM Команда суммирования**

Команда DWSUM суммирует указанное количество (n) 32-битных блоков двоичных данных в операнде, указанном в s. Результат сохраняется в операндах, указанных от d (Array\_d[1]) до d + 3 (Array\_d[4]).

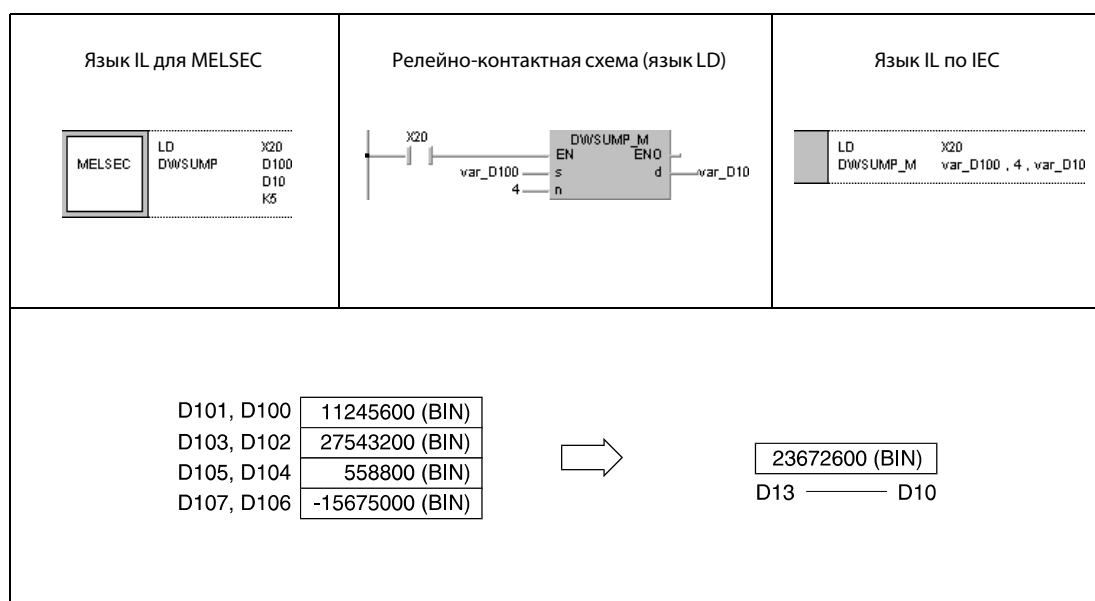
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанная с помощью n область в операнде, указанном в s, находится вне области, предусмотренной для сохранения (код ошибки 4101).

**Пример****DWSUMP**

Следующая программа при положительном фронте сигнала X20 суммирует 32-битные блоки двоичных данных в D100...D107 и сохраняет результат в регистрах D10...D13.

**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

## 7.6 Команды для структурирования программы

Команды для структурирования программы позволяют вызывать программы и области программы или переходить между различными областями программы. Кроме того, к этим командам относятся команды для индексированной адресации и команды повторения.

Следующая таблица содержит обзор всех команд.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту IEC
Команды повторения	FOR	FOR_M
	NEXT	NEXT_M
	BREAK	BREAK_MD
	BREAKP	BREAK_P_MD
Вызов подпрограммы	CALL	CALL_M
	CALLP	CALLP_M
	RET	RET_M
	FCALL	FCALL_MD
	FCALLP	FCALL_P_MD
Вызов подпрограммы в другом программном файле (возможно только в GX Developer)	ECALL	ECALL_M
	ECALLP	ECALLP_M
	EFCALL	EFCALL_M
	EFCALLP	EFCALLP_M
Переключение между областями Main и Sub программы	CHG	CHG_M
Вызов программы микрокомпьютера	SUB	SUB_M
	SUBP	SUBP_M
Индексированная адресация части программы	IX	IX_MD
	IXEND	IXEND_MD
Сохранение индексированных адресов операндов в индексном перечне	IXDEV	IXDEV_M
	IXSET	IXSET_M

7.6.1 FOR, NEXT

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

Операнды														Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9011				
битовые				словные (16 бит)						Константы		Указатели							Уровень			
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1						Z	V	K	H (16#)
n							●	●	●	●	●	●	●	●	●	●	●				3/1	●

<sup>1</sup> Команда FOR расходует три шага, а команда NEXT – один шаг.  
 Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

Операнды MELSEC Q

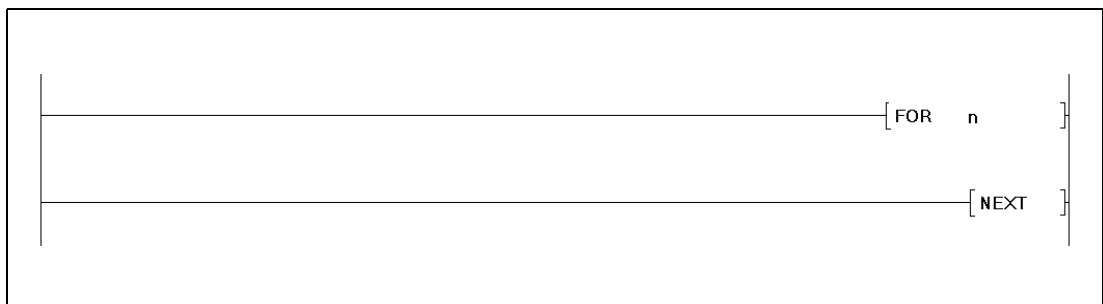
Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные							
n	●	●	●	●	●	●	●	●	—	SM0	2/1

<sup>1</sup> Команда FOR расходует два шага, а команда NEXT – один шаг.

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
n	Количество прогонов цикла FOR-NEXT (от 1 до 32767)	BIN, 16 бит



**Принцип действия****Цикл FOR - NEXT****FOR, NEXT****Комбинация команд**

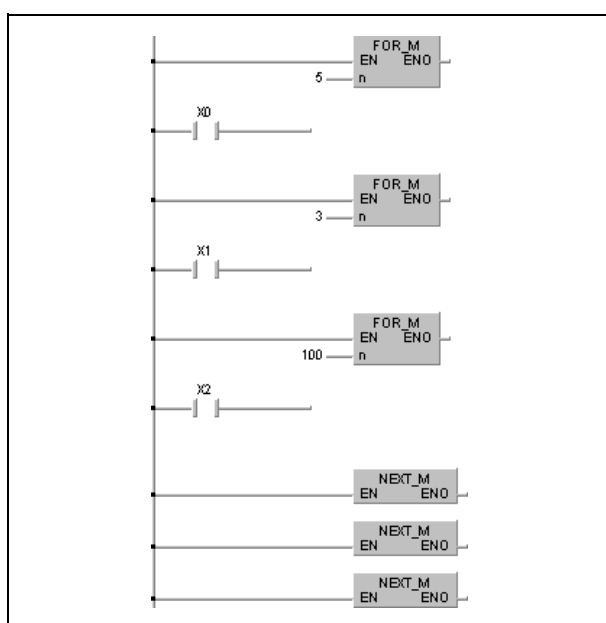
Комбинация команд FOR-NEXT позволяет повторять отдельные фрагменты программы без задания входного условия. При этом повторяется часть программы, заключенная между командами FOR и NEXT. Количество повторений устанавливается в n.

После n-ного выполнения цикла FOR-NEXT выполняется очередной шаг, расположенный за командой NEXT.

Для n можно задать значение между 1 и 32767. Если в n указан 0 или отрицательное значение, n принимается за 1. Таким образом, цикл в любом случае выполняется хотя бы один раз.

Если требуется избежать выполнения фрагмента программы между FOR и NEXT, то его можно пропустить с помощью команды перехода (CJ или SCJ).

В общей сложности можно программировать до 16 уровней вложения команды FOR (для процессоров серии "A" = 5 вложений). Принцип вложения пояснен на следующей иллюстрации.

**Источники ошибок**

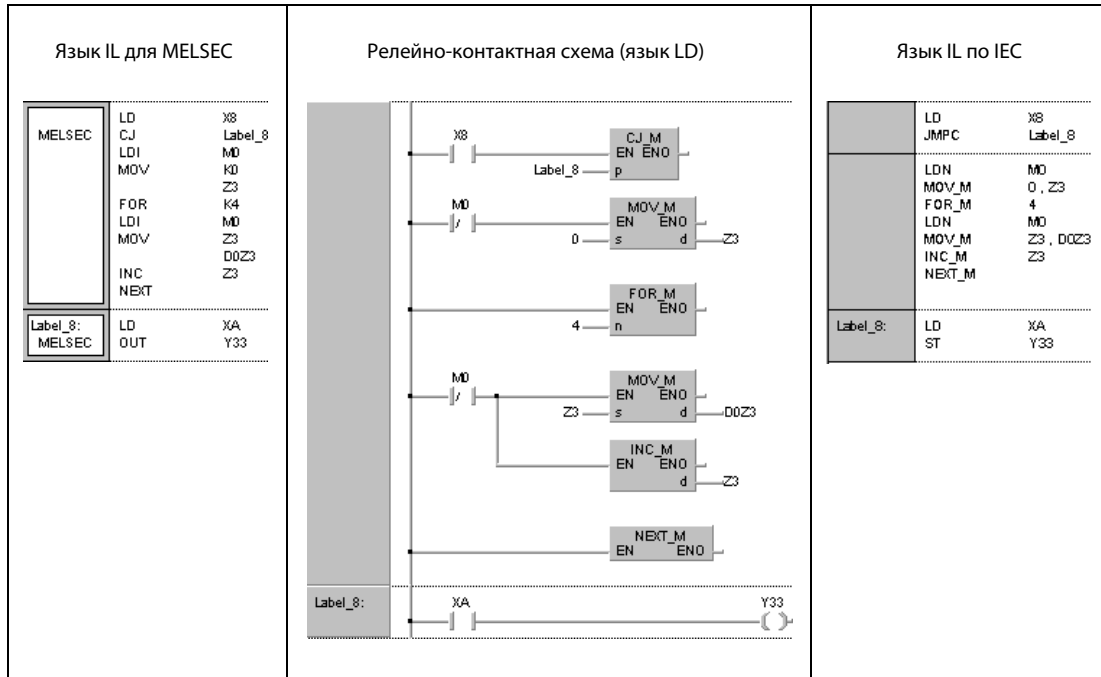
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Команда END/FEND выполняется после выполнения команды FOR, однако перед обработкой команды NEXT (серии "Q"/"System Q" = код ошибки 4200).
- Команда NEXT выполняется перед командой FOR (серии "Q"/"System Q" = код ошибки 4201).
- Количество команд FOR не совпадает с количеством команд NEXT.
- Внутри цикла FOR-NEXT выполняется команда JMP, цель перехода которой расположена за пределами цикла FOR-NEXT.
- Внутри цикла FOR-NEXT имеется команда STOP (серии "Q"/"System Q" = код ошибки 4200).
- Превышена допустимая глубина вложения (серии "Q"/"System Q" = код ошибки 4202).

**ПРИМЕЧАНИЯ** Чтобы прервать выполнение цикла FOR-NEXT до истечения предусмотренного числа прогонов, используется команда BREAK (только для серий "Q" и "System Q").

Чтобы связать команду FOR-NEXT с условием переключения, используйте команду EGP/EGF (только серия "Q").

**Пример** Следующая программа, если X8 не установлен, 4 раза подряд выполняет фрагмент программы между FOR и NEXT. Если X8 установлен, цикл FOR-NEXT пропускается.



### 7.6.2 BREAK, BREAKP

**Процессор**

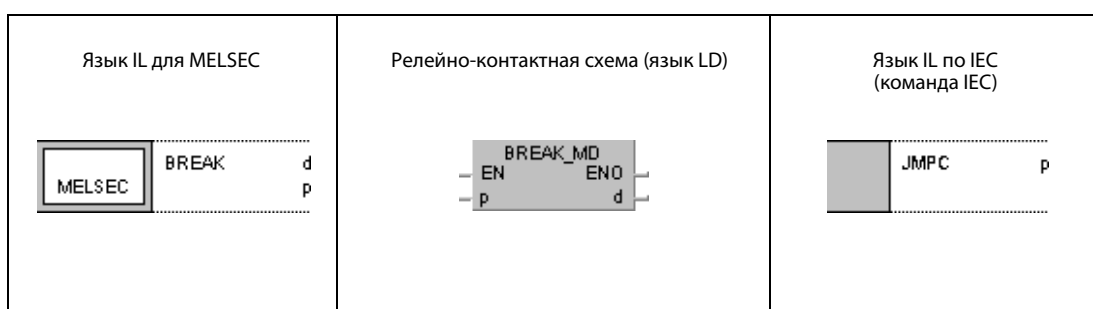
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	●

<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

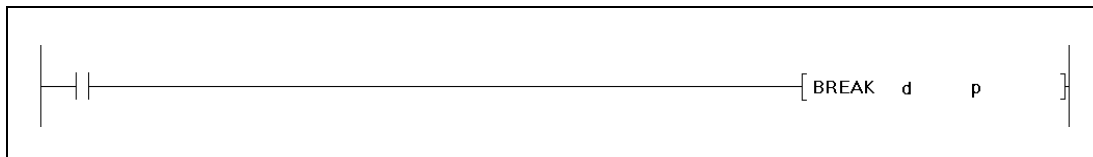
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непор. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные P
	битовые	словные		битовые	словные						
d	●	●	●	●	●	●	—	—	SM0	3	
p	●	●	●	●	●	●	—	●			

**GX IEC Developer**



**GX Developer**

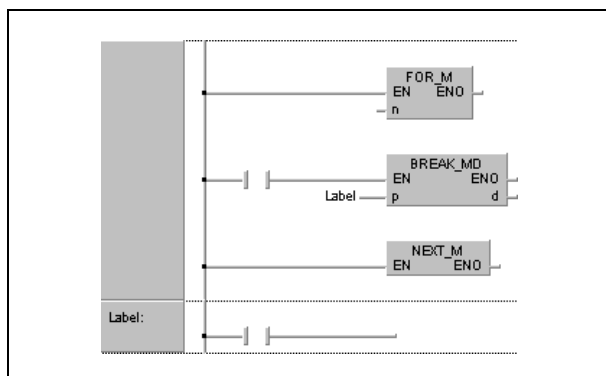


**Переменные**

Операнд	Значение	Тип данных
d	Операнд, в котором сохраняется остающееся количество прогонов цикла FOR-NEXT.	BIN, 16 бит
p	Адрес (указатель/метка), к которому происходит переход после выполнения команды.	указатель/метка

**Принцип действия****Прерывание цикла FOR-NEXT во время его выполнения****BREAK Прерывание цикла FOR-NEXT**

Команда BREAK прерывает цикл FOR-NEXT во время его выполнения и осуществляет переход к указателю (метке), заданному в p.



Количество остающихся циклов FOR-NEXT сохраняется в операнде, указанном в d. Сохраненное в d значение учитывает количество выполненных команд BREAK.

Команду BREAK можно использовать только во время выполнения цикла FOR-NEXT.

Команда BREAK действует только на один уровень вложения. Если имеются несколько вложенных уровней, то для выхода из этих уровней следует запрограммировать соответствующее количество команд BREAK.

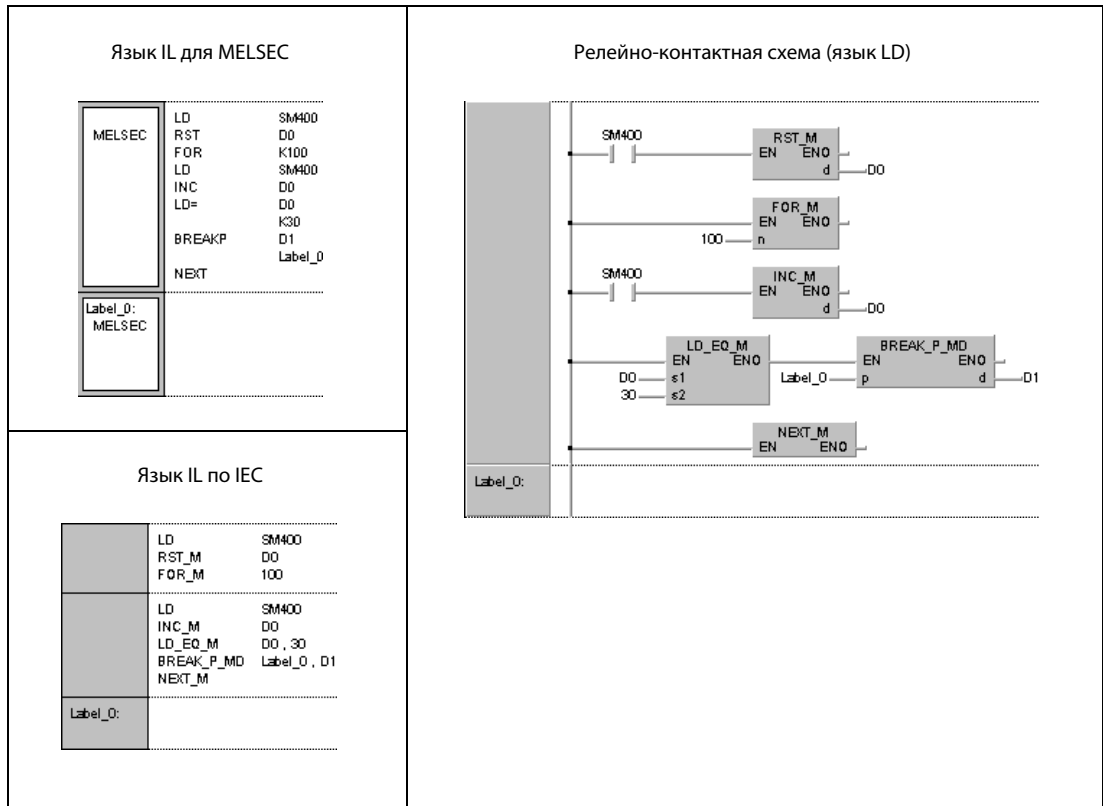
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Команда BREAK использована без цикла FOR-NEXT (код ошибки 4203).
- По адресу, указанному указателем или меткой, подпрограммы не имеется (код ошибки 4210).

**Пример** BREAKP

Следующая программа прерывает выполнение цикла FOR-NEXT при 30-м прогоне и осуществляет переход к части программы, обозначенной меткой Label\_0. Количество оставшихся невыполненными прогонов цикла FOR-NEXT (70) сохраняется в D1.



### 7.6.3 CALL, CALLP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

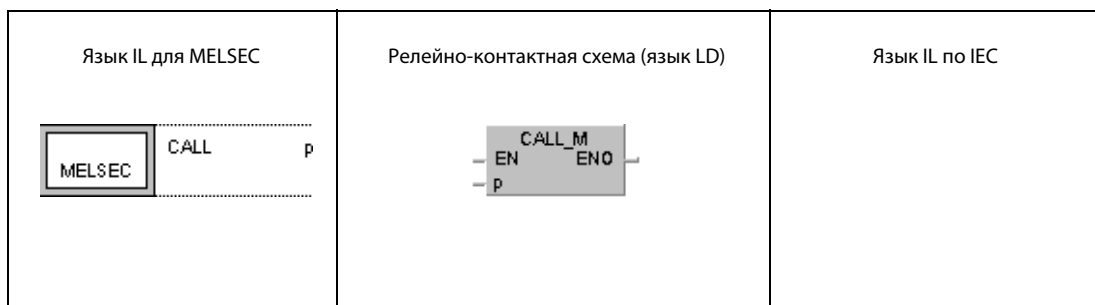
	Операнды															Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки					
	битовые					словные (16 бит)					Константы		Указатели		Уровень				M9012	M9010 M9011					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z				V	K	H (16#)	P	I	N	
p																			●				3 <sup>1</sup>		●

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

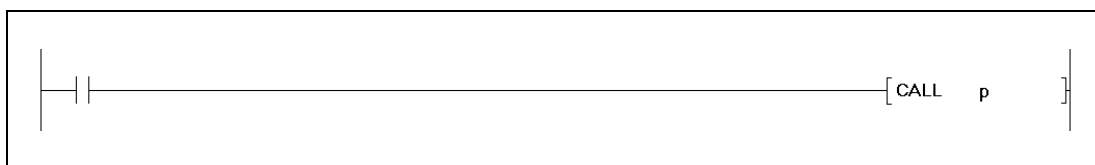
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непр. адр. J□□□		Спец. модули U□\G□	Индексные регистры Zn	Константы			Иные
	битовые	словные		битовые	словные						
p	—	●	●	—	—	—	—	—	—	SM0	2

**GX IEC Developer**



**GX Developer**



**Переменные**

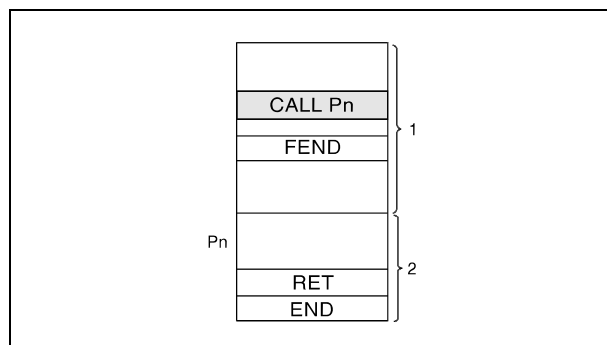
Операнд	Значение	Тип данных
pn	Адрес (указатель/метка) подпрограммы.	указатель/метка

**ПРИМЕЧАНИЕ**

Команду CALL не следует использовать в редакторах по стандарту МЭК (IEC), так как структуру подпрограмм вырабатывает среда программирования GX IEC Developer.

**Принцип действия****Вызов подпрограммы****CALL Вызов подпрограммы**

Команда CALL вызывает подпрограмму по заданному адресу указателя P<sub>xx</sub> подпрограммы в MELSEC MEDOC или по заданной метке подпрограммы в GX IEC Developer. В процессорах серии "A" адрес указателя (метки) может находиться в диапазоне между P(Label)0 и P(Label)255, а в процессорах серий "Q" или "System Q" – в диапазоне между P(Label)0 и P(Label)4095. Адрес указателя (метка) программируется в соответствии с разъяснениями для команд переходов (см. CJ, SCJ, JMP).



<sup>1</sup> основная программа

<sup>2</sup> подпрограмма

Команда CALL вызывает подпрограмму, обозначенную адресом указателя или меткой. В общей сложности при программировании команды CALL допускаются до 5 вложений подпрограмм для серии "A" или до 16 вложений подпрограмм для серий "Q" и "System Q".

Операнды, которые были установлены во время выполнения подпрограммы, остаются установленными даже в том случае, если эта подпрограмма более не выполняется. Для сброса этих операндов следует применять команду FCALL.

**Источники ошибок**

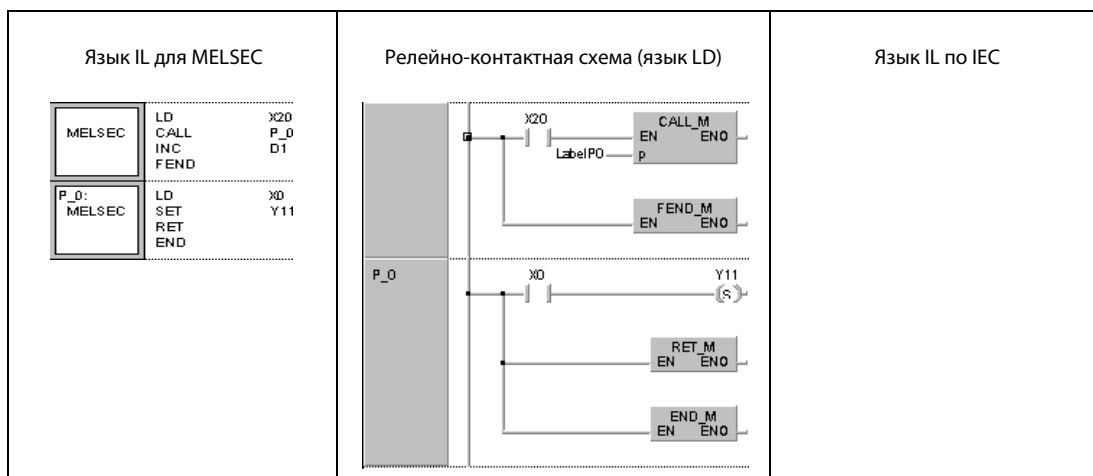
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- После выполнения команды CALL выполняется команда END, FEND, GOEND или STOP, без предшествующей ей команды RET (серии "Q"/"System Q" = код ошибки 4211).
- Перед командой CALL выполняется команда RET (серии "Q"/"System Q" = код ошибки 4212).
- Выполняются более 5 вложений (серия "A") или более 16 вложений (серии "Q"/"System Q") (серии "Q"/"System Q" = код ошибки 4213).
- По адресу, указанному указателем или меткой, подпрограммы не имеется (серии "Q"/"System Q" = код ошибки 4210).
- В команде CALL указан адрес указателя или метка, превышающие P(Label)255 (серия "A").
- Перед выполнением команды RET осуществлен выход из подпрограммы с помощью команды JMP (серия "A").

**Пример**

## CALL

В следующем примере при включенном входе X20 выполняется подпрограмма, находящаяся по указателю (метке) P\_0.

**ПРИМЕЧАНИЯ**

В среде MELSEC команды FEND, END и RET программирует пользователь. После отработки соответствующего программного компонента (POU) никакой другой компонент более не выполняется, так как он находится за командой FEND, END или RET.

Вместо программирования в этой среде можно программировать в стандартном редакторе МЭК (IEC). В этом случае среда программирования GX IEC Developer вставляет команды FEND, END и RET автоматически.



### 7.6.4 RET

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

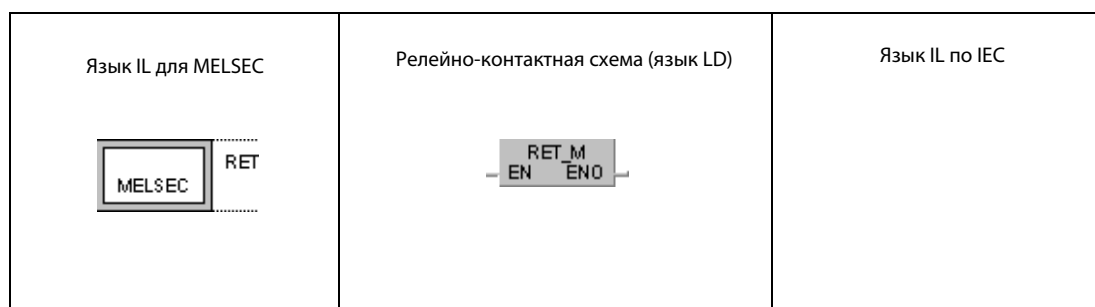
Операнды															Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011				
битовые					словные (16 бит)					Константы		Указатели		Уровень									
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V	K	H (16#)	P
																					1	●	●

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

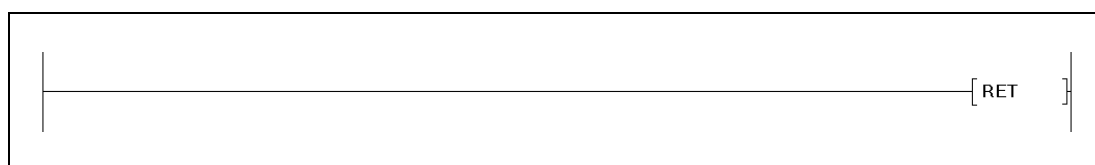
**Операнды MELSEC Q**

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы	Иные			
битовые	словные		битовые	словные							
—	—	—	—	—	—	—	—	—	—	SM0	1

**GX IEC Developer**



**GX Developer**

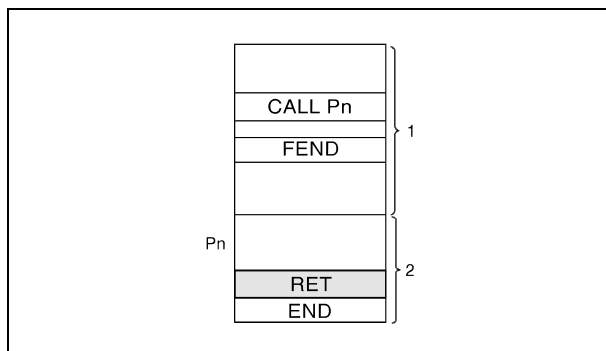


**Переменные**

Операнд	Значение	Тип данных
—	—	—

**Принцип действия****Конец подпрограммы****RET Возврат в основную программу**

Командой RET обозначается конец подпрограммы. Возврат происходит к шагу основной программы, следующему за командой CALL, FCALL, ECALL или EFCALL.



<sup>1</sup> основная программа

<sup>2</sup> подпрограмма

**ПРИМЕЧАНИЯ**

Между командой RET в подпрограмме и командой END в основной программе следует всегда вставлять команду NOP, так как в противном случае центральный процессор неправильно обрабатывает программу (только в случае серии "A").

На языке IL для MELSEC команды FEND, END и RET должен программировать пользователь. После отработки соответствующего программного компонента (POU) никакой другой компонент более не выполняется, так как он находится за командой FEND, END или RET.

Вместо программирования в этой среде можно программировать в стандартном редакторе МЭК (IEC). В этом случае среда GX IEC Developer вставляет команды FEND, END и RET автоматически.

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- После обработки команды CALL и перед выполнением команды RET выполняется команда END, FEND, GOEND или STOP (серии "Q"/"System Q" = код ошибки 4211).
- Перед командой CALL выполняется команда RET (серии "Q"/"System Q" = код ошибки 4212).

**7.6.5 FCALL, FCALLP**

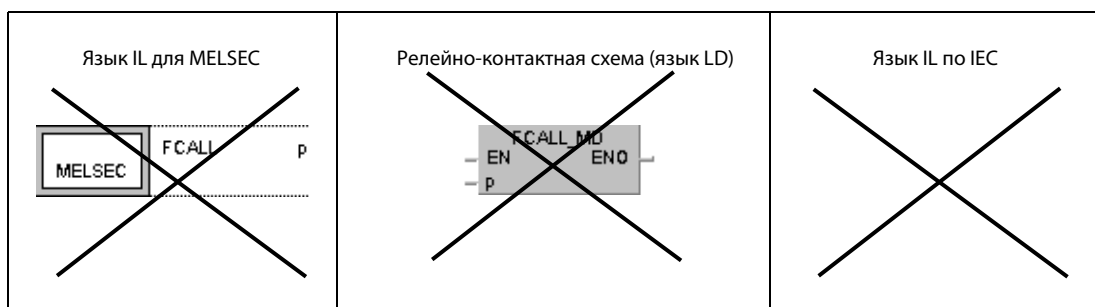
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		●	●	●	●

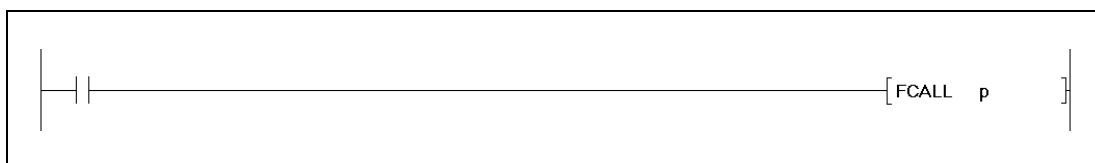
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□NG□	Индексные регистры Zn	Константы	Иные		
	битовые	словные		битовые	словные						
p	—	●	●	—	—	—	—	—	—	SMO	2

**GX IEC Developer**



**GX Developer**



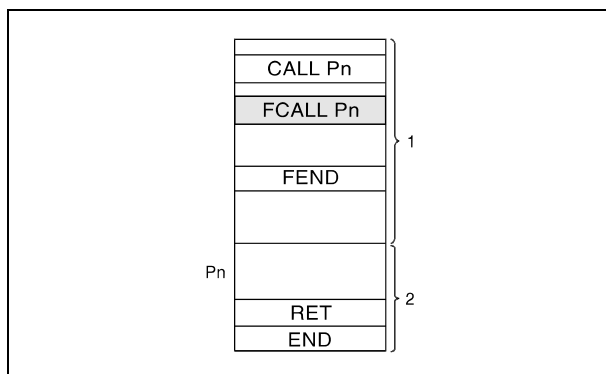
**Переменные**

Операнд	Значение	Тип данных
pn	Адрес (указатель/метка) подпрограммы	указатель/метка

**ПРИМЕЧАНИЕ** В GX IEC Developer эти команды использовать невозможно.

**Принцип действия****Сброс выходов в подпрограммах****FCALL Сброс выходов (в сочетании с командой CALL)**

При сбросе условия выполнения команды FCALL контакты и катушки в подпрограмме, указанной в р (указатель/метка), приводятся в состояние, как если бы условие выполнения команды, управляющей контактом или катушкой, не было установлено.



<sup>1</sup> основная программа

<sup>2</sup> подпрограмма

Ниже указано состояние катушек и контактов после выполнения команды FCALL (т. е. состояние, которое катушки и контакты принимают, если условие выполнения соответствующей команды не установлено):

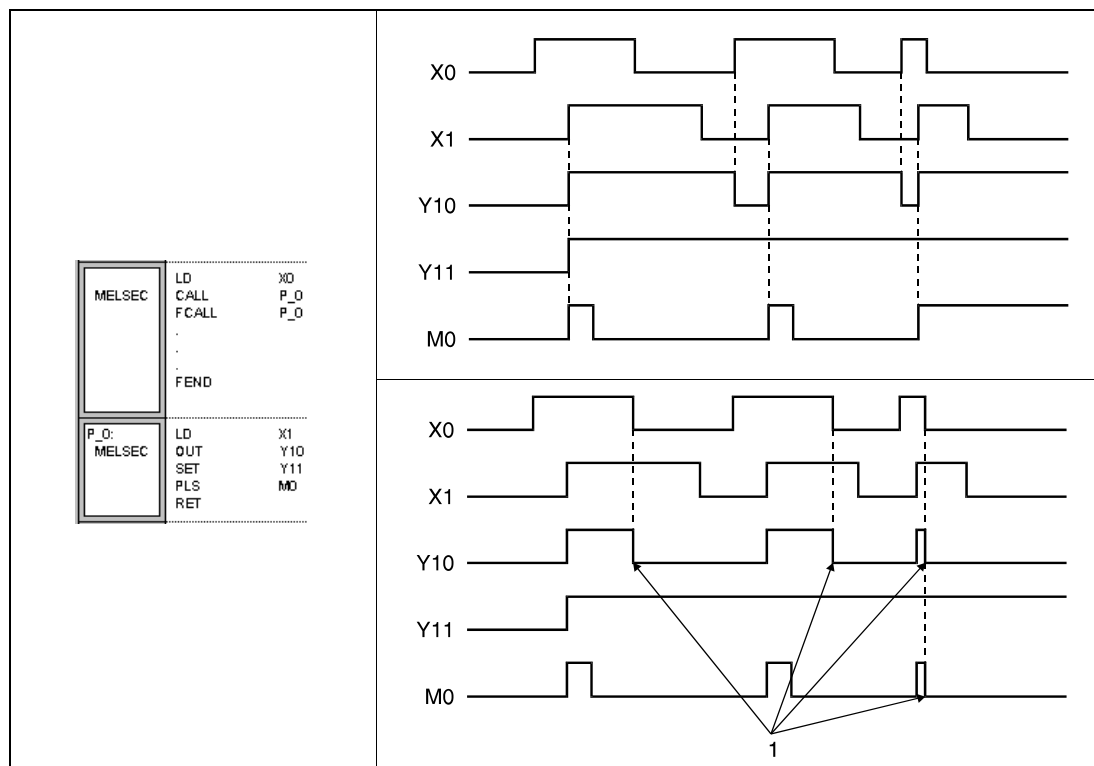
Команды	Состояние контактов и катушек
Команда OUT	Все контакты и катушки, в отношении которых выполняется команда OUT, сбрасываются.
Команда SET	
Команда RST	Контакты и катушки, в отношении которых выполняются эти команды, сохраняют свое состояние.
Команда SFT	
Прикладные команды, часть 1	
Прикладные команды, часть 2	
Команда PLS	Контакты и катушки, в отношении которых выполняются эти команды, принимают состояние, как если бы условия выполнения этих команд не были установлены.
Команды с выработкой выходного импульса	
Настройки медленных и быстрых таймеров	Настройки устанавливаются на 0.
Настройки фиксируемых таймеров	
Настройки счетчиков	
	Настройки сохраняются.

Команда FCALL применяется в сочетании с командой CALL.

На следующей иллюстрации показана программа, применяющая команды CALL и FCALL. Справа на иллюстрациях приведены диаграммы состояния различных контактов, управляемых различными командами. Вверху справа показано состояние этих контактов без применения команды FCALL, а внизу справа – с применением команды FCALL.

Если используется только команда CALL, то после сброса условия выполнения команды CALL состояния управляемых подпрограммой контактов и катушек сохраняются (см. рис. вверху справа).

Если применяется команда FCALL, то после сброса условия выполнения команды FCALL состояния управляемых подпрограммой контактов и катушек сбрасываются (см. рис. внизу справа). То же самое относится к катушкам и контактам, управляемым командами OUT и PLS или командой для выработки импульсов.



<sup>1</sup> В этих точках контакты сбрасываются.

Команда FCALL вызывает подпрограмму, обозначенную адресом указателя или меткой. В общей сложности при программировании команды FCALL допускаются до 16 вложений подпрограмм.

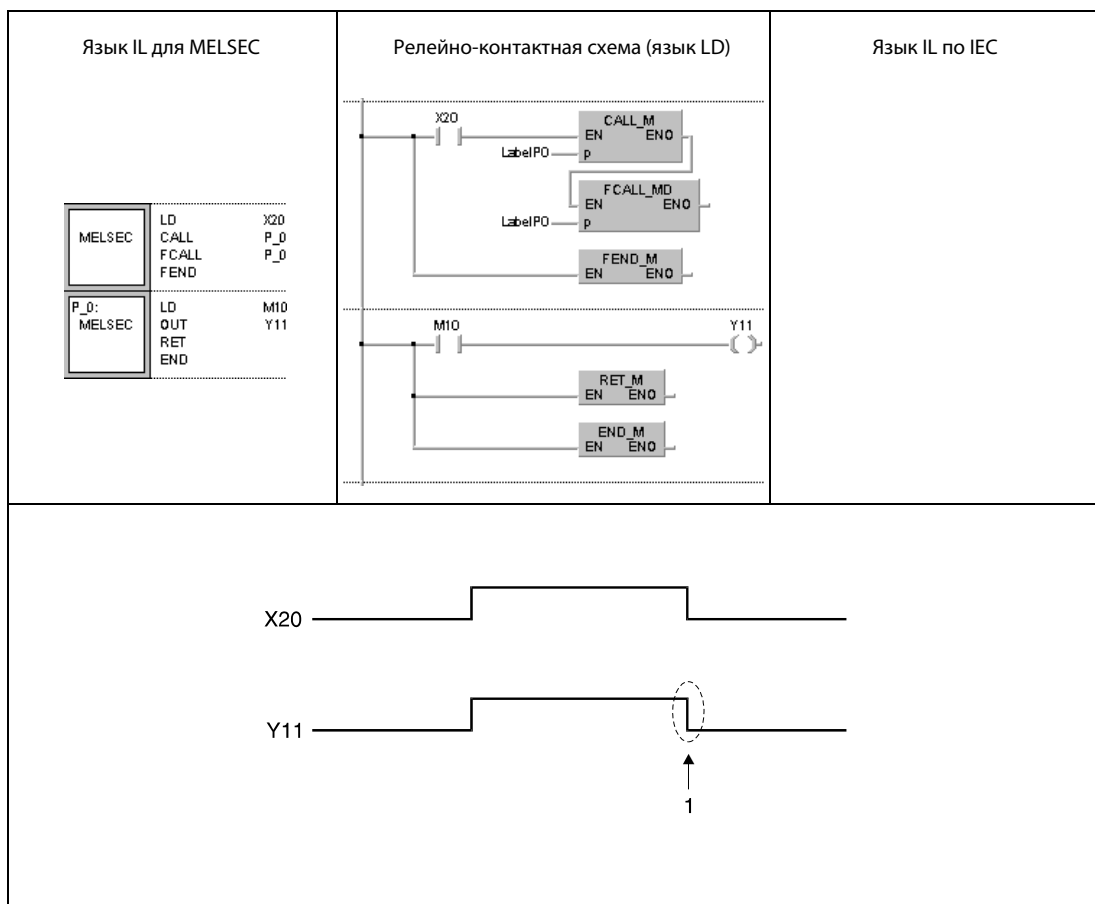
### Источники ошибок

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- После выполнения команды FCALL и перед выполнением команды RET выполняется команда END, FEND, GOEND или STOP (код ошибки 4211).
- Команда RET выполняется перед командой FCALL (код ошибки 4212).
- Выполняются более 16 вложений (код ошибки 4213).
- По адресу, указанному указателем или меткой, подпрограммы не имеется (код ошибки 4210).

**Пример** FCALL

В следующем примере при включенном входе X20 выполняется подпрограмма, находящаяся по адресу указателя (на метке) P\_0. В результате применения команды FCALL, при сбросе X20 сбрасывается и выход Y11 (1).



### 7.6.6 ECALL, ECALLP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□\G□	Индексные регистры Zп	Константы			Иные
	битовые	словные		битовые	словные						
p	—	—	—	—	—	—	—	—	●	SM0	3
s1	● <sup>1</sup>	●	●	●	●	●	●	●	—		
s2	● <sup>1</sup>	●	●	●	●	●	●	●	—		
s3	● <sup>1</sup>	●	●	●	●	●	●	●	—		
s4	● <sup>1</sup>	●	●	●	●	●	●	●	—		
s5	● <sup>1</sup>	●	●	●	●	●	●	●	—		

<sup>1</sup> Маркеры ошибок (F) использовать нельзя.

**GX IEC Developer**

<p>Язык IL для MELSEC</p>	<p>Релейно-контактная схема (язык LD)</p>	<p>Язык IL по IEC</p>
---------------------------	---	-----------------------

**GX Developer**

**Переменные**

Операнд	Значение	Тип данных
Название файла	Название файла подпрограммы, в котором находится вызываемая подпрограмма.	строковая величина
pn	Первый адрес (указатель/метка) подпрограммы.	указатель/метка
от s1 до s5	Операнды, передаваемые подпрограмме	бит BIN, 16 бит BIN, 32 бита

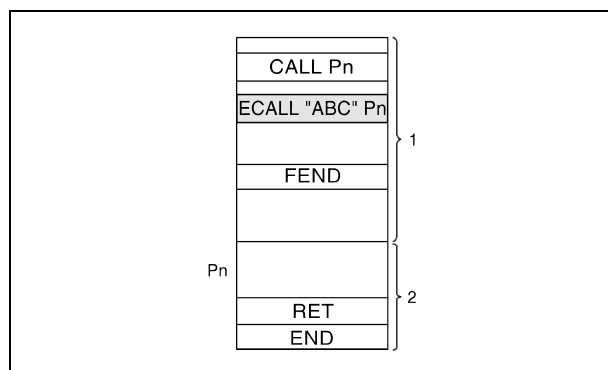
**ПРИМЕЧАНИЕ** В GX IEC Developer эти команды использовать невозможно.

**Принцип действия**

**Вызов подпрограммы в программном файле**

**ECALL Вызов подпрограммы**

Для вызова подпрограммы в программном файле с помощью команды ECALL указывается название программного файла и метка подпрограммы. Адрес указателя (метка) может находиться в диапазоне между P(Label)0 и P(Label)4095. Адрес указателя (метка) программируется в соответствии с разъяснениями для команд переходов (см. CJ, SCJ, JMP).



<sup>1</sup> основная программа (файл "Main")

<sup>2</sup> подпрограмма (файл "ABC")

В качестве названия файла можно указывать только названия программных файлов, хранящихся во внутренней памяти (дисконд 0).

При вызове программных файлов расширение названия файла указывать не требуется.

Команда ECALL вызывает подпрограмму, обозначенную адресом указателя или меткой. В общей сложности при программировании команды ECALL допускаются до 16 вложений подпрограмм.

Операнды, которые были установлены во время выполнения подпрограммы, остаются установленными даже в том случае, если эта подпрограмма более не выполняется. Для сброса этих операндов следует применять команду EFCALL.

Если подпрограмма использует функциональные входы, выходы или регистры (FX, FY и FD), то операнды можно передавать с помощью переменных s1...s5. Перед выполнением подпрограммы битовые операнды передаются на функциональные входы, а словные операнды – в функциональные регистры. После обработки подпрограммы состояния или содержимое FY и FD передаются в указанные операнды.

Количество данных, которое может принять функциональный регистр, зависит от операндов, указанных в s1...s5. В случае констант, индексных регистров и побиточно адресованных битовых операндов они могут принять максимум 2 слова, в случае словных операндов – максимум 4 слова. Например, если в s2 указан операнд D0, в FD1 сохраняются D0, D1, D2 и D3.

Количество используемых подпрограммой функциональных входов, выходов и регистров должно совпадать с количеством операндов, переданных в s1...s5 при вызове команды ECALL.

Тип функциональных операндов должен совпадать с типом операндов, переданных командой ECALL.

Операнды, указанные в s1...s5, не должны перекрываться.



**Источники ошибок**

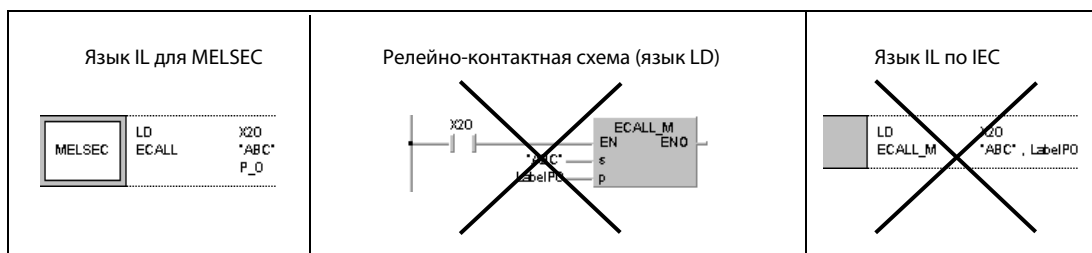
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- После выполнения команды ECALL и перед выполнением команды RET выполняется команда END, FEND, GOEND или STOP (код ошибки 4211).
- Команда RET выполняется перед командой ECALL (код ошибки 4212).
- Выполняются более 16 вложений (код ошибки 4213).
- По адресу, указанному указателем или меткой, подпрограммы не имеется (код ошибки 4210).
- В s1...s5 указан функциональный вход (FX), функциональный выход (FY) или функциональный регистр (код ошибки 4101).
- Указанный программный файл не существует (код ошибки 4210).
- Указанный программный файл не может быть выполнен (код ошибки 2411).

**Пример**

## ECALL

В следующем примере при включенном входе X20 выполняется подпрограмма, расположенная на указателе (метке) P\_0 в программном файле "ABC".



7.6.7 EFCALL, EFCALLP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>


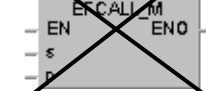
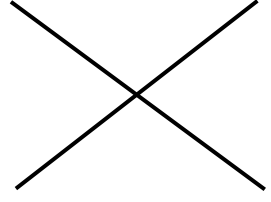
<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

Операнды MELSEC Q

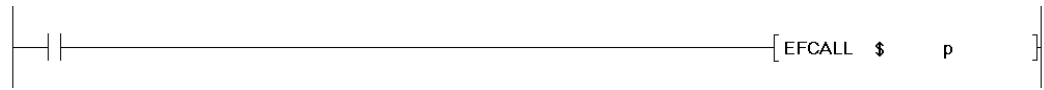
	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□□G□□	Индексные регистры Zn	Константы			Иные
	битовые	словные		битовые	словные						
p	—	—	—	—	—	—	—	—	●	SM0	3
s1	● <sup>1</sup>	●	●	●	●	●	●	●	—		
s2	● <sup>1</sup>	●	●	●	●	●	●	●	—		
s3	● <sup>1</sup>	●	●	●	●	●	●	●	—		
s4	● <sup>1</sup>	●	●	●	●	●	●	●	—		
s5	● <sup>1</sup>	●	●	●	●	●	●	●	—		

<sup>1</sup> Маркеры ошибок (F) использовать нельзя.

GX IEC Developer

<p>Язык IL для MELSEC</p> 	<p>Релейно-контактная схема (язык LD)</p> 	<p>Язык IL по IEC</p> 
---	---	---

GX Developer



Переменные

Операнд	Значение	Тип данных
Название файла	Название файла подпрограммы, в котором находится вызываемая подпрограмма.	строковая величина
pn	Адрес (указатель/метка) подпрограммы	указатель/метка
от s1 до s5	Операнды, передаваемые подпрограмме.	бит BIN, 16 бит BIN, 32 бита

**ПРИМЕЧАНИЕ** В GX IEC Developer эти команды использовать невозможно.

**Принцип действия**

**Сброс выходов в подпрограммах, находящихся в программных файлах**

**EFCALL                      Сброс выходов (в сочетании с командой ECALL)**

При сбросе условия выполнения команды EFCALL контакты и катушки подпрограммы, расположенной по адресу р (указатель/метка) в файле с указанным названием, приводятся в состояние, как если бы условие выполнения команды, управляющей контактом или катушкой, не было установлено.

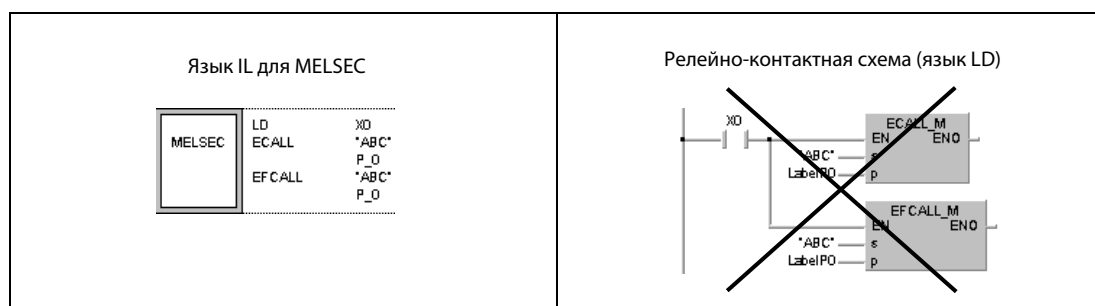
Команда EFCALL выполняет подпрограммы, находящиеся в ином программном файле чем тот, из которого они вызываются.

Ниже указано состояние катушек и контактов после выполнения команды EFCALL (т. е. состояние, которые принимают катушки и контакты, если условие выполнения управляющей ими команды не установлено):

Команды	Состояние контактов и катушек
Команда OUT	Все контакты и катушки, в отношении которых выполняется команда OUT, сбрасываются.
Команда SET	Контакты и катушки, в отношении которых выполняются эти команды, сохраняют свое состояние.
Команда RST	
Команда SFT	
Прикладные команды, часть 1	
Прикладные команды, часть 2	
Команда PLS	Контакты и катушки, в отношении которых выполняются эти команды, принимают состояние, как если бы условия выполнения этих команд не были установлены.
Команды с выработкой выходного импульса	Настройки устанавливаются на 0.
Настройки медленных и быстрых таймеров	
Настройки фиксируемых таймеров	
Настройки счетчиков	Настройки сохраняются.

Команда EFCALL применяется в сочетании с командой ECALL.

На следующей иллюстрации показана программа, применяющая команды ECALL и EFCALL.



Если подпрограмма использует функциональные входы, выходы или регистры (FX, FY и FD), то операнды можно передавать с помощью переменных s1...s5. Перед выполнением подпрограммы битовые операнды передаются на функциональные входы, а словные операнды – в функциональные регистры. После обработки подпрограммы состояния или содержимое FY и FD передаются в указанные операнды.

Количество данных, которое может принять функциональный регистр, зависит от операндов, указанных в s1...s5. В случае констант, индексных регистров и поблочко адресованных битовых операндов они могут принять максимум 2 слова, в случае словных операндов – максимум 4 слова. Например, если в s2 указан операнд D0, в FD1 сохраняются D0, D1, D2 и D3.

Количество используемых подпрограммой функциональных входов, выходов и регистров должно совпадать с количеством операндов, передаваемых в s1...s5 при вызове команды EFCALL.

Тип функциональных операндов должен совпадать с типом операндов, передаваемых командой EFCALL.

Операнды, указанные в s1...s5, не должны перекрываться.

Команда EFCALL вызывает подпрограмму, обозначенную адресом указателя (меткой). В общей сложности при программировании команды EFCALL допускаются до 16 вложений подпрограмм.

**Источники ошибок**

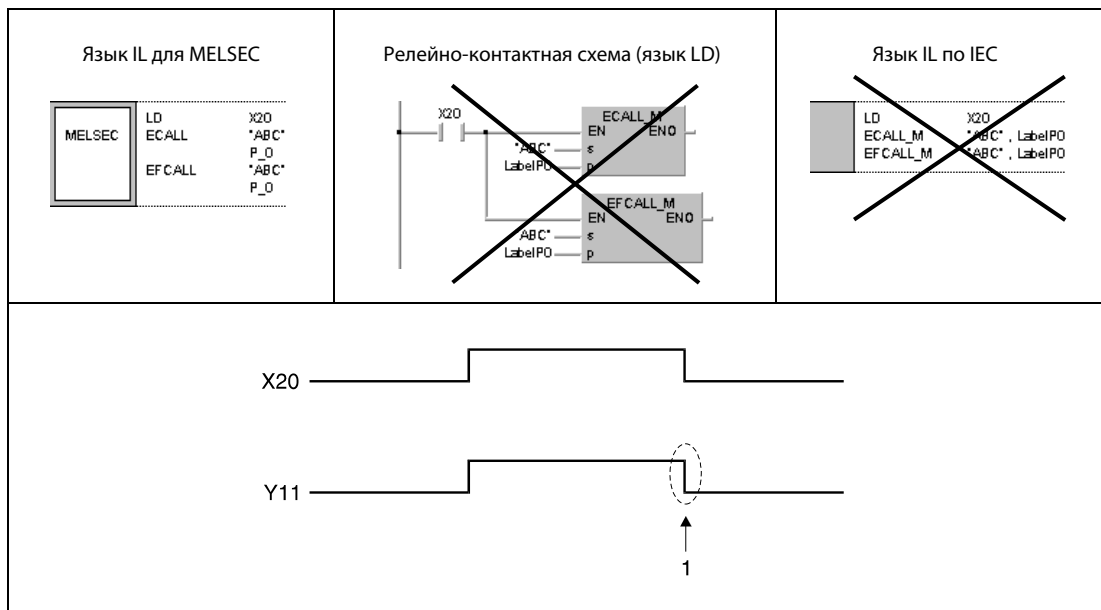
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- После выполнения команды EFCALL и перед выполнением команды RET выполняется команда END, FEND, GOEND или STOP (код ошибки 4211).
- Перед командой EFCALL выполняется команда RET (код ошибки 4212).
- Выполняются более 16 вложений (код ошибки 4213).
- По адресу, указанному указателем или меткой, подпрограммы не имеется (код ошибки 4210).
- В s1, s2, s3, s4 или s5 указан маркер ошибки (F) (код ошибки 4101).
- Указанный программный файл не существует (код ошибки 4210).
- Указанный программный файл не может быть выполнен (код ошибки 2411).

**Пример**

EFCALL

В следующем примере при включенном входе X20 выполняется подпрограмма, находящаяся по адресу указателя (на метке) P\_0 в программном файле "ABC". В результате применения команды EFCALL, при сбросе X20 сбрасывается и выход Y11 (1).



7.6.8 CHG

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
	● <sup>1</sup>	● <sup>2</sup>	● <sup>3</sup>		

<sup>1</sup> только процессоры A3N

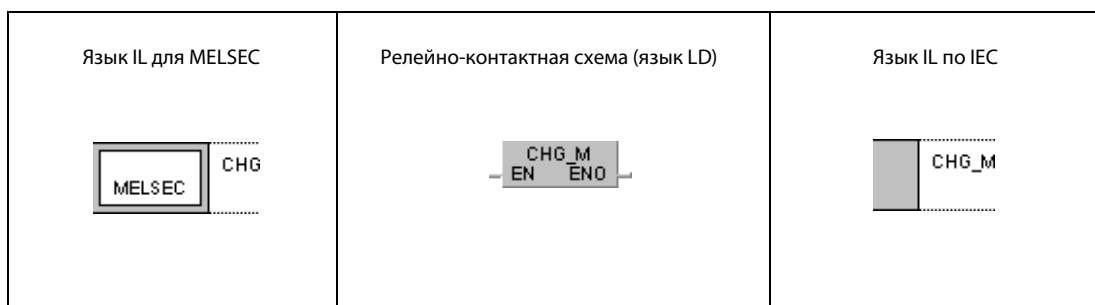
<sup>2</sup> только процессоры A3A

<sup>3</sup> только процессоры A3U

Операнды MELSEC A

Операнды																	Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011		
битовые							словные (16 бит)							Константы		Указатели						Уровень	
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K						H (16#)	P
																					1		

GX IEC Developer

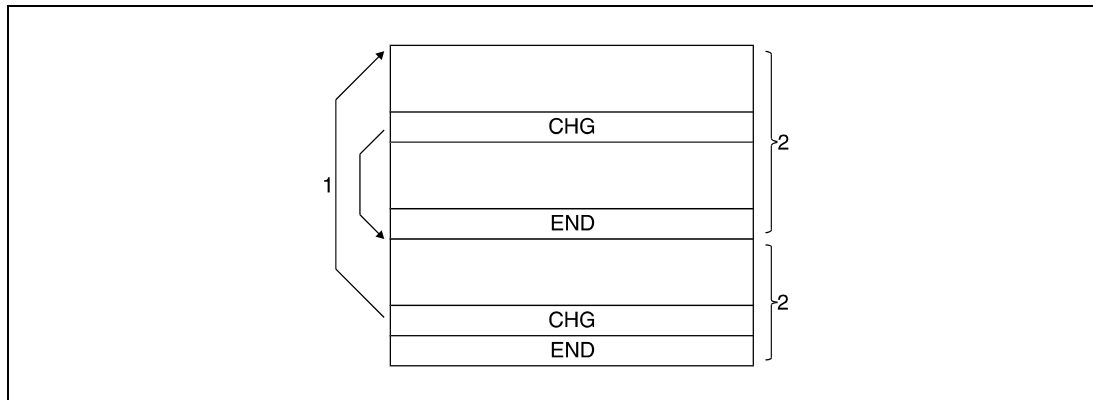


Переменные

Операнд	Значение	Тип данных
—	—	—

**Принцип  
действия****Переключение между областями MAIN и SUB программы****CHG      Команда переключения**

При установленном входном условии команда CHG переключает программу между областями MAIN и SUB. Переключение происходит после обработки таймеров и счетчиков, а также после самодиагностики.



<sup>1</sup> обработка таймеров и счетчиков, самодиагностика

<sup>2</sup> программа

## Переключение между областями MAIN и SUB программы

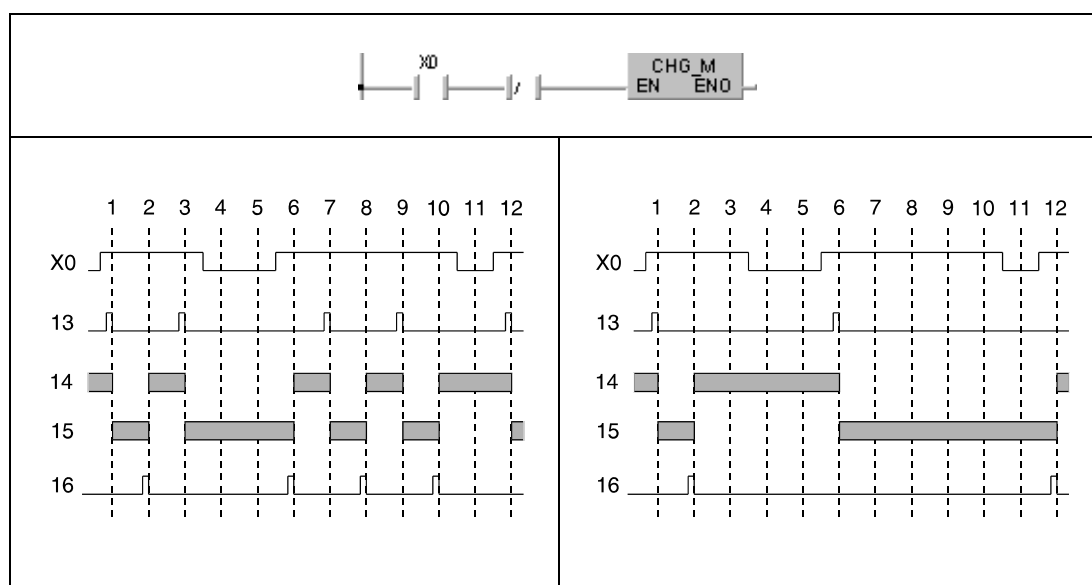
### CHG В случае процессора A3□

В процессоре A3□ команда CHG выполняется только при положительном фронте входного условия. Результат обработки входного условия зависит от состояния специального маркера M9050. Таким образом, при изменении состояния маркера M9050 изменяется функция команды CHG.

Специальный маркер M9050 не применим для процессоров A3N. Процессор A3N ведет себя так, как если бы маркер M9050 был установлен.

В верхней части следующей иллюстрации показано применение команды CHG. Эта часть программы находится перед командой END или FEND в области MAIN или SUB.

В нижней части отражены соответствующие изменения сигналов. Левая диаграмма сигналов относится к неустановленному маркеру M9050, а правая – к установленному маркеру M9050. Таблица под иллюстрацией поясняет обработку в зависимости от состояния X0.



Строка 13 показывает выполнение команды CHG в области MAIN, строка 14 – область MAIN, строка 15 – область SUB, а строка 16 – выполнение команды CHG в области SUB.

Состояние X0	Состояние M9050	
	не установлен	установлен
0	Без переключения между областями MAIN и SUB памяти (4, 5, 11).	Без переключения между областями MAIN и SUB памяти (4, 5, 11).
1	Команда CHG выполняется в каждом программном цикле и переключает программу между областями MAIN и SUB (2, 3, 7, 8, 9, 10).	Переключение из MAIN в SUB и снова в MAIN происходит только при первом положительном фронте X0 (2).
0 → 1	Переключение между областями MAIN и SUB памяти (1, 6, 12).	Переключение между областями MAIN и SUB памяти (1, 6, 12).

После выполнения команды CHG происходит обработка команды END в текущей программе. Обработка начинается с шага 0 другой программы. В конце области MAIN и SUB среда GX IEC Developer выполняет переключение автоматически.



**Команда CHG в сочетании с командой PLS**

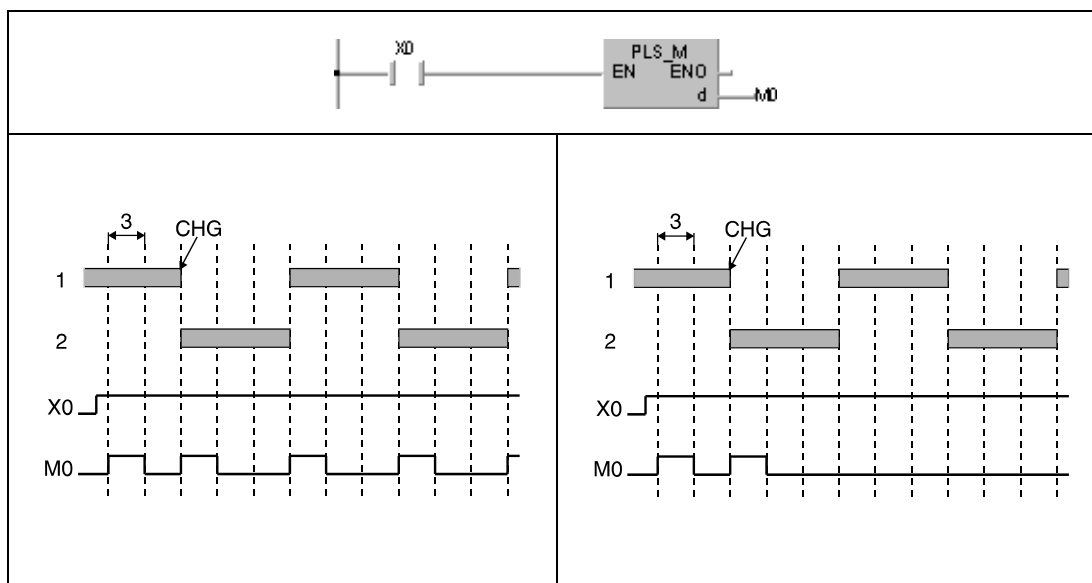
**CHG В случае процессора A3□**

Если программа используется для процессора A3□, то характер выполнения команды PLS зависит от состояния специального маркера M9050.

Специальный маркер M9050 не применим для процессоров A3N. Процессор A3N ведет себя так, как если бы маркер M9050 был установлен.

В верхней части следующей иллюстрации показано применение команды PLS. Эта часть программы находится в начале программы (шаг 0), в области памяти MAIN или SUB.

В нижней части отражены соответствующие изменения сигналов. Левая диаграмма сигналов относится к неустановленному маркеру M9050, а правая – к установленному маркеру M9050. Таблица под иллюстрацией поясняет обработку в зависимости от состояния X0.



1 обозначает обработку области MAIN, 2 – обработку области SUB, а 3 – длительность цикла.

Состояние X0	Состояние M9050	
	не установлен	установлен
0	M0 не устанавливается.	M0 не устанавливается.
1	M0 устанавливается только во время первого цикла после переключения программы с помощью команды CHG.	После включения X0 маркер M0 устанавливается только во время первого цикла программы в области SUB, переключение на которую было выполнено с помощью команды CHG.
0 → 1	M0 устанавливается только на один программный цикл.	M0 устанавливается только на один программный цикл.

**Команда CHG в сочетании с импульсной командой**

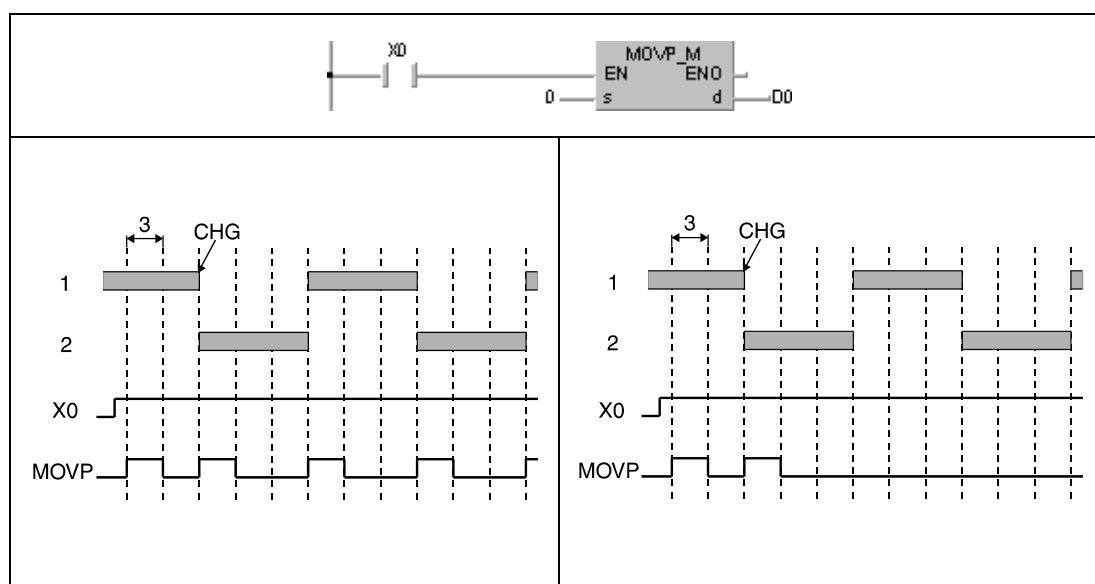
**CHG В случае процессора A3**

Если программа используется для процессора A3, характер выполнения импульсной команды зависит от состояния специального маркера M9050.

Специальный маркер M9050 не применим для процессоров A3N. Процессор A3N ведет себя так, как если бы маркер M9050 был установлен.

В верхней части следующей иллюстрации показано применение импульсной команды. Эта часть программы находится в начале программы (шаг 0), в области памяти MAIN или SUB.

В нижней части отражены соответствующие изменения сигналов. Левая диаграмма сигналов относится к неустановленному маркеру M9050, а правая – к установленному маркеру M9050. Таблица под иллюстрацией поясняет обработку в зависимости от состояния X0.



1 обозначает обработку области MAIN, 2 – обработку области SUB, а 3 – длительность цикла.

Состояние X0	Состояние M9050	
	не установлен	установлен
0	Команда MOV_P не выполняется.	Команда MOV_P не выполняется.
1	Команда MOV_P выполняется только во время первого цикла после переключения программы с помощью команды CHG.	После включения X0 команда MOV_P выполняется только во время первого цикла программы в области SUB, переключение на которую было выполнено с помощью команды CHG.
0 → 1	Команда MOV_P выполняется один раз.	Команда MOV_P выполняется один раз.

**Команда CHG и принцип работы счетчиков**

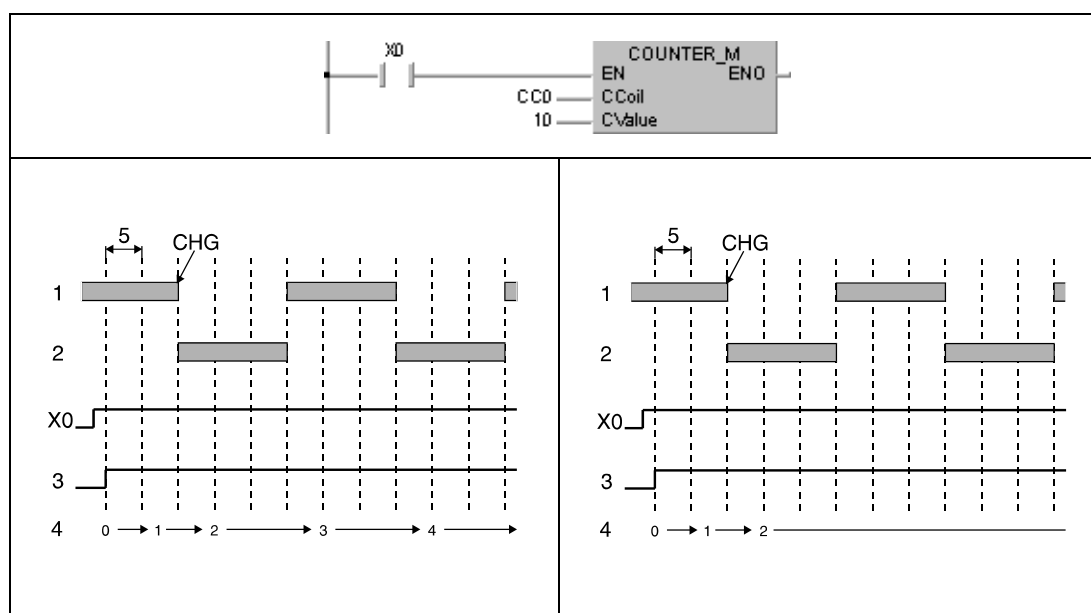
**CHG В случае процессора A3□**

Если программа используется для процессора A3□, принцип работы счетчиков зависит от состояния специального маркера M9050 (если все прочие входные условия остаются без изменений).

Специальный маркер M9050 не применим для процессоров A3N. Процессор A3N ведет себя так, как если бы маркер M9050 был установлен.

В верхней части следующей иллюстрации показано применение команды счетчика. Эта часть программы находится в начале программы (шаг 0), в области памяти MAIN или SUB.

В нижней части отражены соответствующие изменения сигналов. Левая диаграмма сигналов относится к неустановленному маркеру M9050, а правая – к установленному маркеру M9050. Таблица под иллюстрацией поясняет обработку в зависимости от состояния X0.



Строка 1 показывает обработку области MAIN, строка 2 – обработку области SUB, строка 3 – контакт счетчика C0, строка 4 – фактическое значение счетчика C0, а строка 5 – длительность цикла.

Состояние X0	Состояние M9050	
	не установлен	установлен
0	Фактическое значение счетчика не изменяется.	Фактическое значение счетчика не изменяется.
1	Фактическое значение счетчика повышается на 1 после команды END (FEND, CHG) во время первого цикла после переключения программы с помощью команды CHG.	После включения X0 фактическое значение счетчика повышается на 1 после команды END (FEND, CHG) во время первого цикла программы в области SUB, переключение на которую было выполнено с помощью команды CHG.
0 → 1	После выполнения команды END (FEND, CHG) фактическое значение счетчика повышается на 1.	После выполнения команды END (FEND, CHG) фактическое значение счетчика повышается на 1.

### Команда CHG и время таймеров

Все процессоры, способные обрабатывать команду CHG, имеют две области памяти для настроек таймеров. При этом одна область памяти зарезервирована для области MAIN, а одна для области SUB. Это означает, что таймеры обрабатываются только в соответствии с той частью программы, которая выполняется в данный момент (в области MAIN или SUB).

Настройки тех таймеров, которые в данный момент не обрабатываются, в соответствующих областях памяти установлены на "0". Настройка "0" соответствует бесконечно большой настройке, при которой время таймера никогда не истекает.

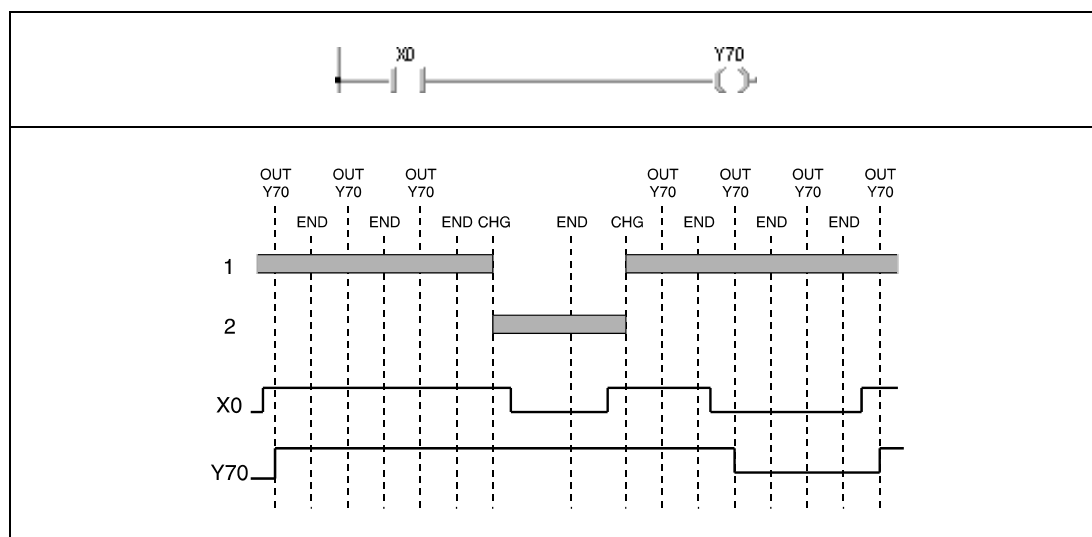
Если после запуска таймера в текущей программе выполнена команда CHG для переключения в область MAIN или SUB, то в той части программы, в которую произошло переключение (активной части программы), этот таймер более не обрабатывается. Это связано с тем, что таймер был запрограммирован в приостановленной программе, и с точки зрения активной программы его настройка интерпретируется как "0". После повторного переключения в первоначальную программу обработка таймера продолжается. Время таймера считается истекшим, если фактическое значение больше настройки или меньше 0. При истекшем времени таймера включается контакт таймера.

### Команда CHG и обработка команд OUT

Процессоры, способные обрабатывать команду CHG, переключают выходные контакты в зависимости от выполняемой в данный момент части программы.

После переключения с одной части программы на другую (MAIN/SUB) выходные контакты сохраняют свое состояние. Состояние сохраняется даже в том случае, если изменяется входное условие.

В верхней части следующей иллюстрации показано применение команды OUT. Эта часть программы находится в области MAIN памяти. В области SUB памяти выход Y70 не применяется.



В нижней части отражены соответствующие изменения сигналов. Числом 1 обозначена обработка области MAIN, а числом 2 – обработка области SUB.

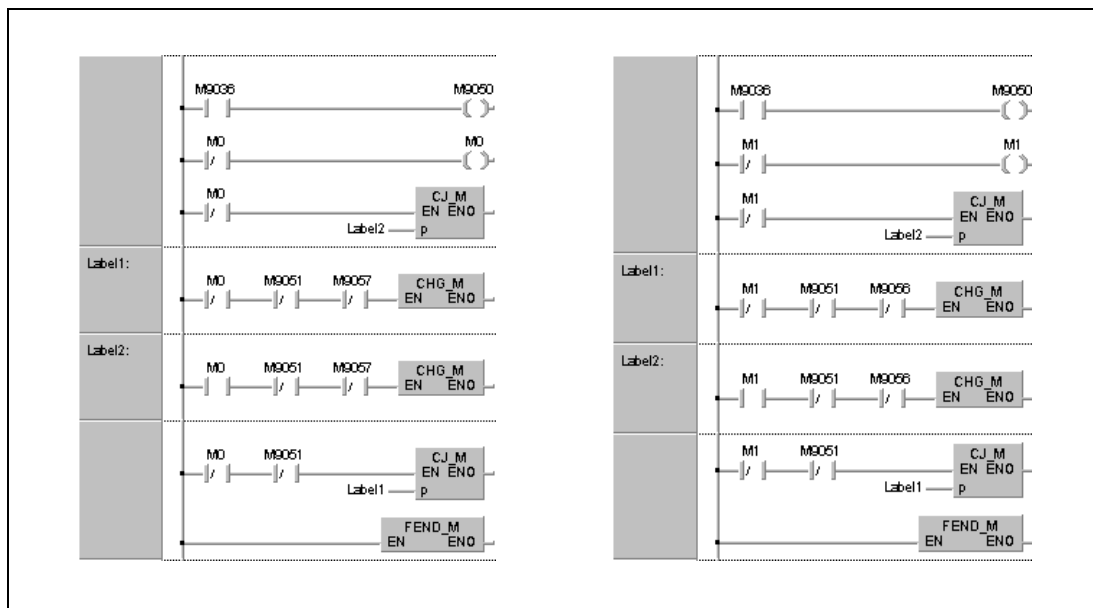
Во время отработки области MAIN выход Y70 включается и выключается в зависимости от входного условия X0. Во время отработки области SUB состояние выхода Y70 остается неизменным даже в том случае, если изменилось входное условие.

**Пример 1** CHG (процессоры A3□)

Для безупречной обработки команды CHG результат обработки программного цикла необходимо сравнивать с результатом предшествующего цикла. Для этого перед выполнением команды CHG следует всегда устанавливать специальный маркер M9050, чтобы загружать результат обработки предшествующего цикла из промежуточной памяти в оперативную память.

Так как в процессоре A3□ команда CHG выполняется только при установленном входном условии, программы следует составлять так, как это показано на следующей схеме. Левый пример программы находится в области Main памяти, а правый пример – в области Sub.

Специальный маркер M9036 установлен всегда.

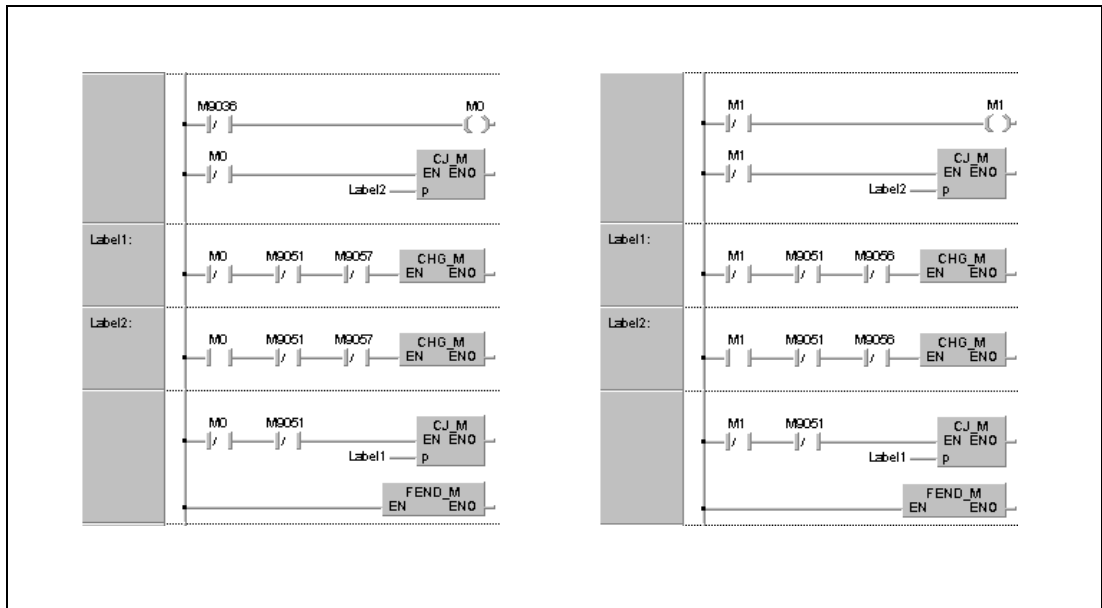


**ПРИМЕЧАНИЯ** Если во время обработки программы в области MAIN изменяется программа в области SUB (или наоборот), то необходимо использовать специальные маркеры M9051, M9056 и M9057, чтобы предотвратить выполнение команды CHG и, тем самым, переключение программы.

Это означает, что если в области SUB выполняются изменения в режиме онлайн, область MAIN не обрабатывается. В GX Developer и GX IEC Developer это происходит автоматически.

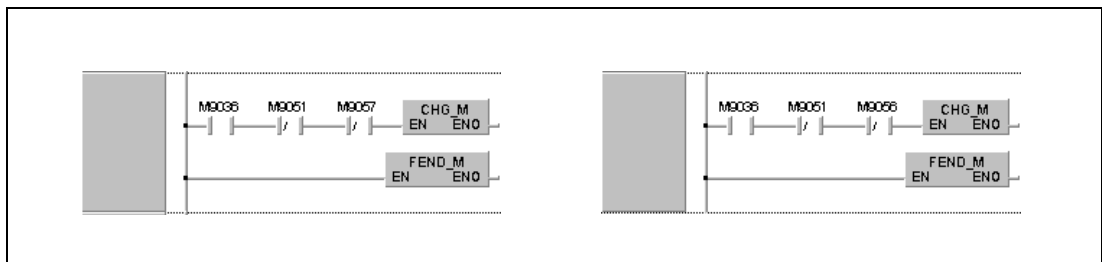
**Пример 2** CHG (процессор A3N)

В процессоре A3N команда CHG выполняется только при установленном входном условии. Программы должны соответствовать схеме в этом примере. Левый пример программы находится в области Main памяти, а правый пример – в области Sub.



**Пример 3** CHG (процессор A3N)

Программы должны соответствовать схеме в этом примере. Левый пример программы находится в области Main памяти, а правый пример – в области Sub.



**7.6.9 SUB, SUBP**

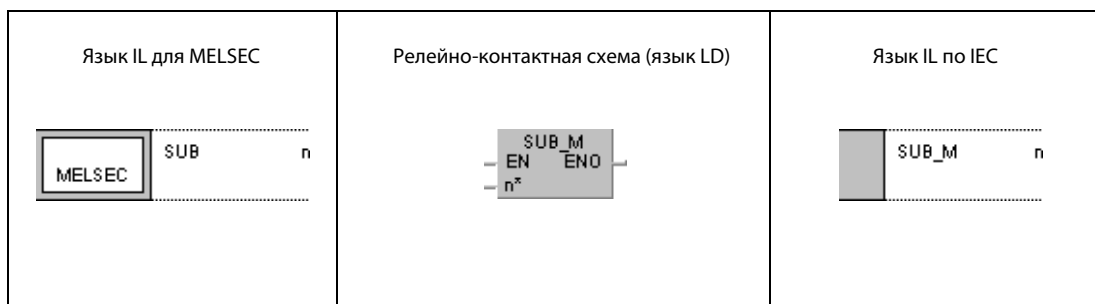
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●				

**Операнды MELSEC A**

Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011					
битовые						словные (16 бит)						Константы		Указатели							Уровень				
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						K	H (16#)	P	I	N
n							●	●	●	●	●	●	●	●	●	●						3	●		●

**GX IEC Developer**



**Переменные**

Операнд	Значение	Тип данных
n	Адрес вызываемой программы микрокомпьютера.	адрес

**Принцип действия**

**Вызов программы микрокомпьютера**

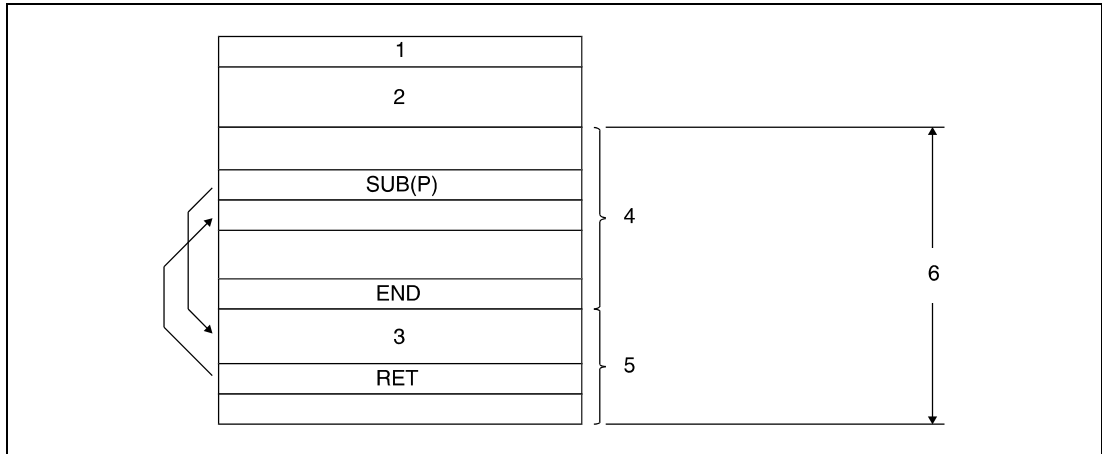
**SUB Вызов программы микрокомпьютера**

Команда SUB/SUBP вызывает пользовательскую программу микрокомпьютера.

При выполненном входном условии команда SUB вызывает программу микрокомпьютера, находящуюся по адресу "n".

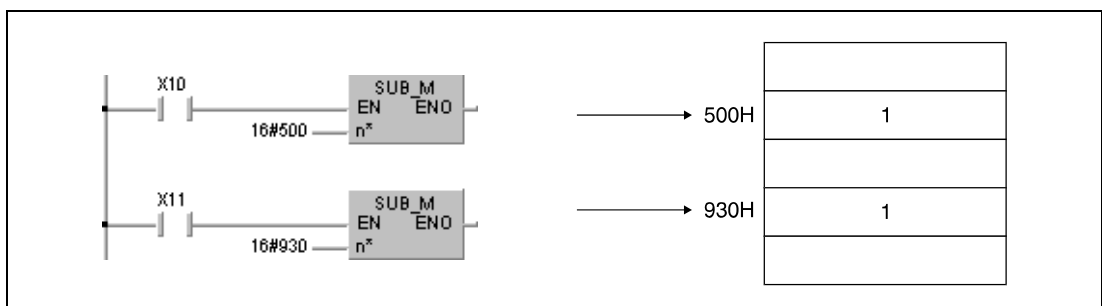
После отработки программы микрокомпьютера основная программа продолжается с шага, следующего за командой SUB/SUBP.

Команду SUB/SUBP можно использовать в областях MAIN и SUB памяти.



- 1 параметры
- 2 настройки таймеров и счетчиков
- 3 программа микрокомпьютера
- 4 область основной программы
- 5 область программы микрокомпьютера
- 6 область MAIN или SUB памяти

В области для программ микрокомпьютера можно создать несколько программ.



- 1 программа микрокомпьютера



**ПРИМЕЧАНИЕ** *В специализированных прикладных командах процессоров AnA, AnAS и AnU (Dedicated Instructions) функция команды SUB заключается в установке 16-битной константы в блоке команд.*

Более подробную информацию о программах микрокомпьютеров см. в разделе 10 этого руководства.

**Источники ошибок** В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Превышена максимальная емкость программы микрокомпьютера.
- Указанный в *n* адрес находится вне допустимого диапазона.

**ПРИМЕЧАНИЯ** *Время обработки программы микрокомпьютера, вызванной с помощью команды SUB(P), не должно превышать 5 мс. Если обработка длится дольше 5 мс, возникает конфликт с основной программой, в результате которого контроллер более не может работать безупречно.*

*Если требуется выполнить программу микрокомпьютера, длящуюся дольше 5 мс, следует разбить эту программу на отдельные блоки и поочередно выполнять эти блоки. Так можно сократить время обработки программы микрокомпьютера.*

## 7.6.10 IX, IXEND

## Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	●

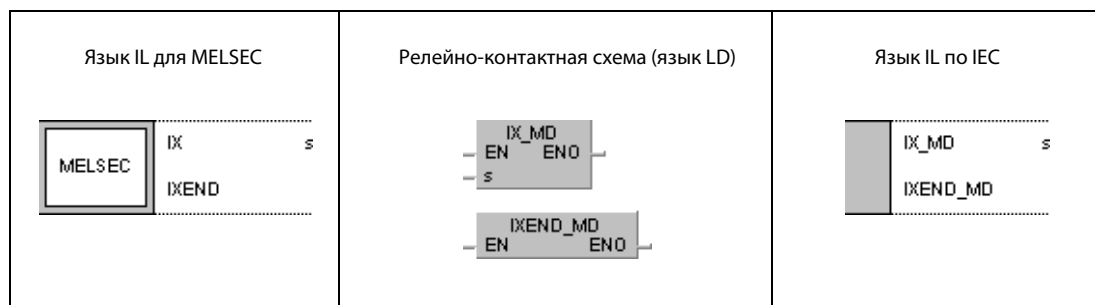
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

## Операнды MELSEC Q

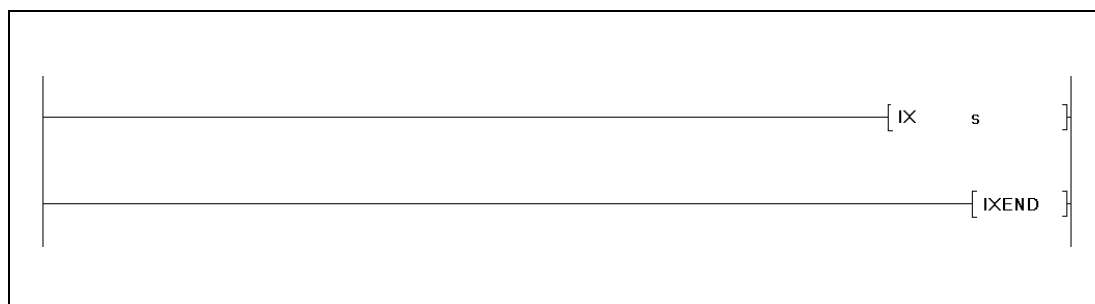
	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	—	SMD	2/1 ● <sup>1</sup>

<sup>1</sup> Команда IX расходует два шага, а команда IXEND – один шаг.

## GX IEC Developer



## GX Developer



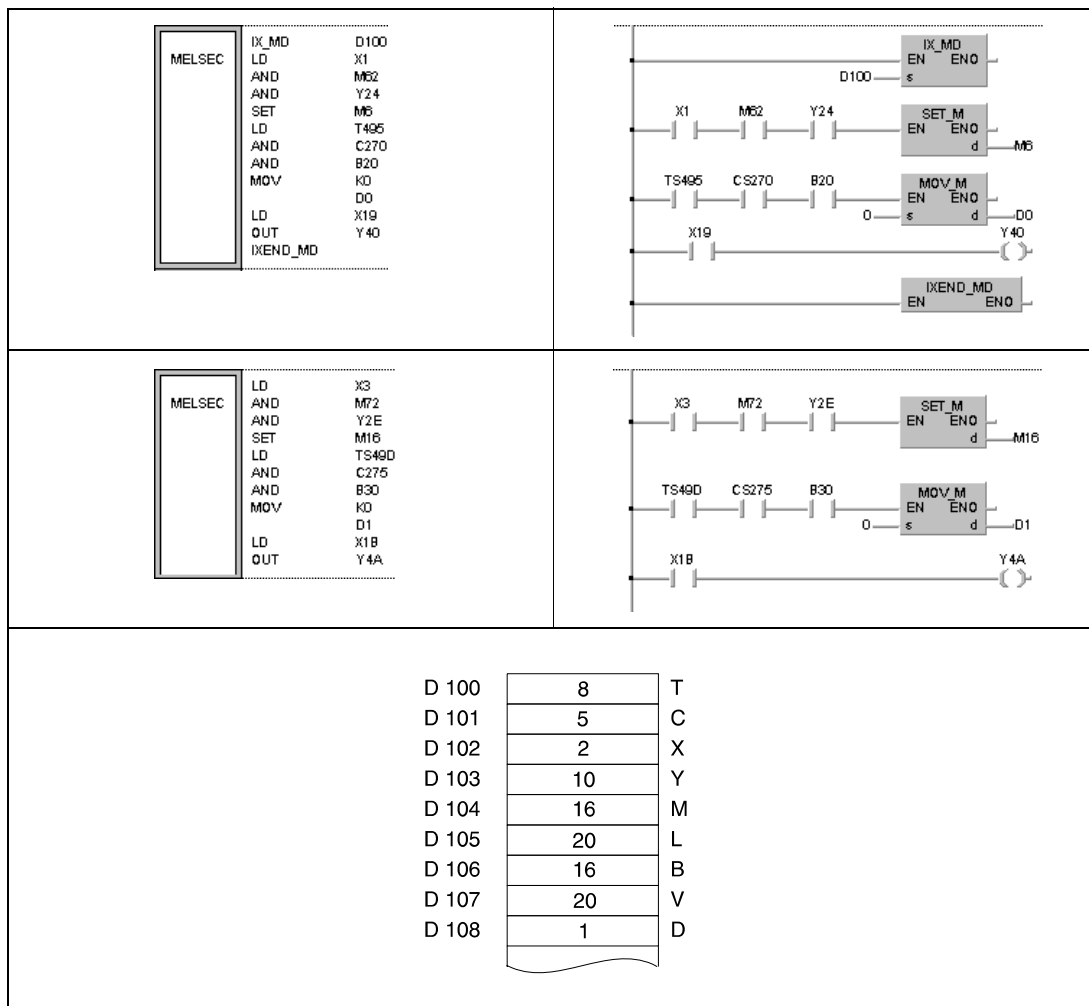
## Переменные

Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранены данные для индексной адресации.	BIN, 16 бит

**Принцип действия** **Индексированная адресация части программы**  
**IX, IXEND** **Команда адресации**

Команды IX и IXEND можно использовать только в режиме MELSEC среды GX IEC Developer. Команды IX и IXEND выполняют индексацию адресов операндов в той части программы, которая находится между командами IX и IXEND.

Во время этого процесса к адресам операндов присоединяются десятичные числовые значения индексного перечня (s). При дальнейшей работе программы действительным адресом является новый адрес в шестнадцатеричном виде. Каждому операнду, указанному в s, сопоставлен определенный тип операнда. К адресу операнда применяется операция суммирования. Индексация пояснена на следующей иллюстрации.



К адресу таймера TS495 прибавляется значение в D100 (8). Новым адресом является TS49D.  
 К адресу счетчика CS270 прибавляется значение в D101 (5). Новым адресом является CS275.  
 К адресам входов X1 и X19 прибавляется значение в D102 (2). Новыми адресами являются X3 и X1B.  
 К адресам выходов Y24 и Y40 прибавляется значение в D103 (10). Новыми адресами являются Y2E и Y4A.

К адресам маркеров M6 и M62 прибавляется значение в D104 (16). Новыми адресами являются M16 и M72.

К адресу маркера связи B20 прибавляется значение в D106 (16). Новым адресом является B30.

К адресу регистра D0 прибавляется значение в D108 (1). Новым адресом является D1.

Индексацию адресов с помощью цикла команд IX/IXEND нельзя применять к командам PLS, PLF и импульсным командам, которые при установленном входном условии выполняются только один раз.

Если новый адрес, полученный путем сложения, выходит за допустимый диапазон адресов, безупречная обработка команды более не обеспечивается.

Если команды IX и IXEND выполняются во время перехода между фрагментами программы в онлайн-режиме (изменения программы во время режима RUN), безупречная обработка также не обеспечивается.

Значения, прибавляемые к адресам словных операндов, в которых возможно обращение к каждому биту, хранятся в виде двоичных данных. Начальный адрес операндов, для которых указываются эти значения, сохранен в s.

Индексная адресация между командами IX и IXEND программы не возможна.

При расширении программы индексированные адреса операндов части программы, заключенной между командами IX и IXEND, преобразуются в адреса, применяющие индексные регистры (Zn). В следующей таблице показана взаимосвязь между индексированными адресами и соответствующими индексными регистрами.

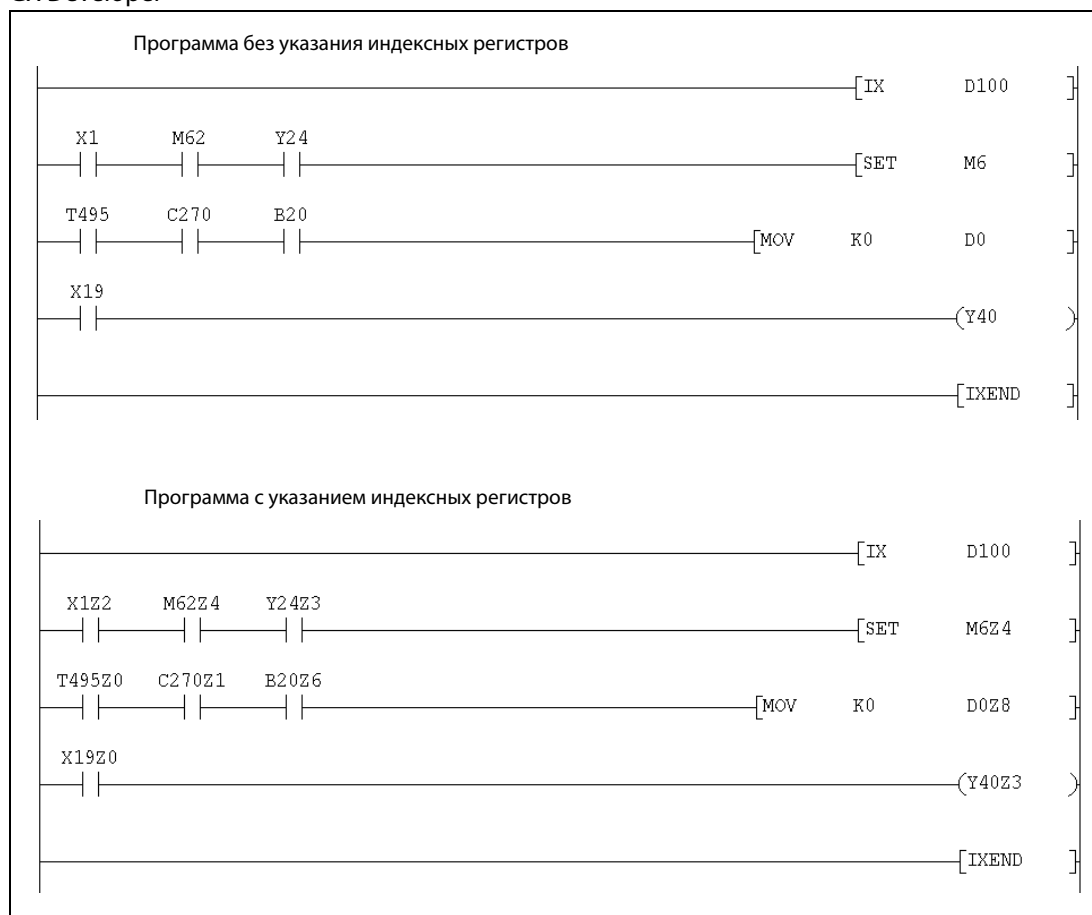
s	Адрес	Индексный регистр	s	Адрес	Индексный регистр
s	Добавочные величины для адресов таймеров (T)	Z0	s + 8	Добавочные величины для адресов регистров данных	Z8
s + 1	Добавочные величины для адресов счетчиков (C)	Z1	s + 9	Добавочные величины для адресов регистров связи (W)	Z9
s + 2	Добавочные величины для адресов входов (X)	Z2	s + 10	Добавочные величины для адресов регистров файлов (R)	Z10
s + 3	Добавочные величины для адресов выходов (Y)	Z3	s + 11	Добавочные величины для адресов ввода-вывода буферных регистров (U)	Z11
s + 4	Добавочные величины для адресов маркеров	Z4	s + 12	Добавочные величины для адресов буферных регистров (G)	Z12
s + 5	Добавочные величины для адресов фиксируемых маркеров (L)	Z5	s + 13	Добавочные величины для сетевых адресов операндов связи с непосредственным доступом (J)	Z13
s + 6	Добавочные величины для адресов маркеров связи (B)	Z6	s + 14	Добавочные величины для локальных адресов регистров файлов (ZR)	Z14
s + 7	Добавочные величины для адресов маркеров фронта (V)	Z7	s + 15	Добавочные величины для адресов указателей (меток)	Z15

В процессорах Q00J, Q00 или Q01 отсутствуют индексные регистры от Z10 до Z15.

В зависимости от используемой среды программирования, может понадобиться в части программы между командами IX и IXEND указать индексные регистры.

**Пример**

GX Developer



Индексные регистры, используемые между командами IX и IXEND (от Z0 до Z15), не влияют на другие команды программы, адресуемые путем индексации.

**ПРИМЕЧАНИЯ** Для индексной адресации операндов части программы необходимо запустить периферийные устройства в стандартном режиме и выполнить расширение программы (только для серии QnA).

Если периферийные устройства запускаются процессорами Q2A, Q2A-S1, Q3A или Q4A и одновременно применяется индексированная адресация операндов части программы между командами IX и IXEND, безупречная обработка не обеспечивается.

Предусмотрите блокировку, предотвращающую одновременное выполнение нескольких команд IX и IXEND, если в обычной программе и в программе прерывания используются команды IX и IXEND. В обычной программе заблокируйте прерывания между командами IX и IXEND.

**Источники ошибок**

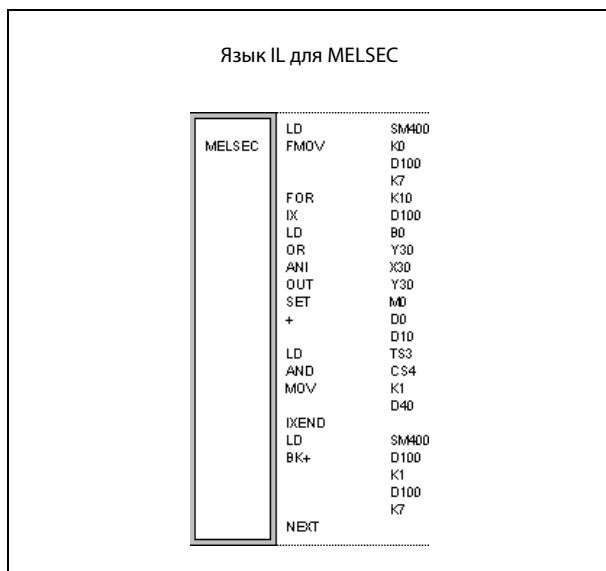
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Команды IX и IXEND не запрограммированы в виде пары команд (код ошибки 4231).
- После выполнения команды IX и перед выполнением команды IXEND выполняется команда END, FEND, GOEND или STOP (код ошибки 4231).

## Пример

## IX, IXEND

Следующая программа проходит цикл между командами IX и IXEND в общей сложности 10 раз. При каждом прогоне адреса запрограммированных в этой области операндов повышаются на 1. В таблице под иллюстрацией перечислены регистры, в которых содержатся добавляемые значения для соответствующих операндов. Кроме того, показано изменение адресов операндов для 1-го, 2-го, 3-го и 10-го прогона цикла.



D	Добавочные величины	Изменение адресов операндов в отдельных прогонах			
		1.	2.	3.	10.
D100	Добавочные величины для адресов таймеров (T)	T3	T4	T5	TC
D101	Добавочные величины для адресов счетчиков (C)	C4	C5	C6	CD
D102	Добавочные величины для адресов входов (X)	X10	X11	X12	X19
D103	Добавочные величины для адресов выходов (Y)	Y30	Y31	Y32	Y39
D104	Добавочные величины для адресов маркеров (M)	M0	M1	M2	M9
D106	Добавочные величины для адресов маркеров связи (B)	B0	B1	B2	B9
D108	Добавочные величины для адресов регистров данных	D0	D1	D2	D9
		D10	D11	D12	D19
		D40	D41	D42	D49

### 7.6.11 IXDEV, IXSET

**Процессор**

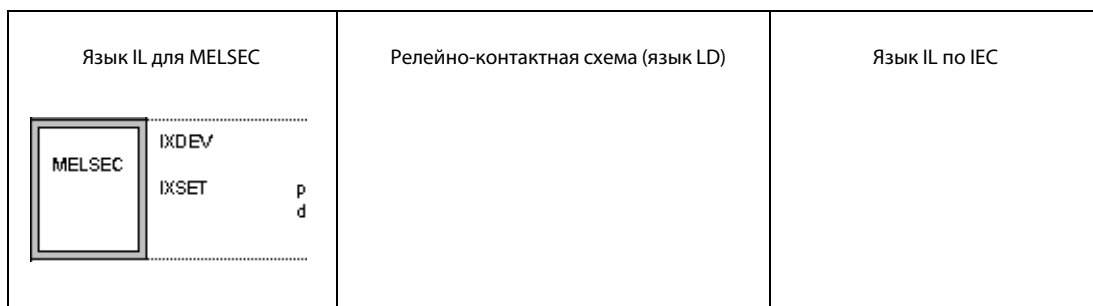
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)			Иные P
	битовые	словные		битовые	словные						
p	—	—	—	—	—	—	—	—	●	SM0	1/3 ● <sup>1</sup>
d	—	●	●	—	—	—	—	—	—		

<sup>1</sup> Команда IXDEV расходует один шаг, а команда IXSET – три шага.

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
p	Первый адрес операнда (только указатель/метка), в котором сохранены данные для индексной адресации.	указатель/метка
d	Первый адрес операнда, в котором сохраняются индексированные адреса операндов.	BIN, 16 бит

**Принцип действия****Сохранение индексированных адресов операндов в индексном перечне****IXDEV/IXSET Команда записи в индексный перечень**

Команды IXDEV и IXSET можно использовать только в режиме MELSEC среды GX IEC Developer или в среде GX Developer.

Команды IXDEV и IXSET считывают адреса операндов, расположенных в области смещения, и записывают эти величины смещения в виде индексного перечня в операнд, указанный в d.

Взаимосвязь между типами операндов и соответствующими регистрами указана в разделе, в котором описаны команды IX и IXEND.

Если в области смещения нет операнда какого-либо типа, то в индексный перечень для этого операнда записывается 0.

Отдельные биты словных операндов обрабатываются в качестве фиктивного контакта, т. е. считывание и занесение в индексный перечень возможно только для адреса отдельного бита. При адресации фиктивного контакта указывается номер соответствующего бита. Например, бит 0 (b0) в регистре данных D0 указывается в виде D0.0. Для указания бита в 16-битном слове данных используются шестнадцатеричные цифры от 0 до F.

Значения смещения считываются следующим образом:

- Считывание операндов: T□, C□, X□, Y□, M□, L□, V□, B□

Считывается и записывается в соответствующий регистр величина смещения, обозначенная знаком □.

- Считывание операндов: D□.XX, W□.XX, R□.XX<sup>1</sup>, U□\G□.XX<sup>1</sup>, ZR□.XX<sup>1</sup>

Считывается и записывается в соответствующий регистр величина смещения, обозначенная знаком □.

Символами XX обозначена переменная для указания бита.

<sup>1</sup>кроме процессоров Q00J, Q00 и Q01

- Считывание операндов: J□\B□<sup>1</sup>, J□\W□<sup>1</sup>, J□\X□<sup>1</sup>, J□\Y□<sup>1</sup>

Считывается и записывается в соответствующий регистр величина смещения, обозначенная знаком □.

Если для операнда, следующего за J□\, не требуется записывать смещение, то соответствующее значение следует установить на 0.

<sup>1</sup>Кроме процессоров Q00J, Q00 и Q01

- При программировании команды IXSET смещение операнда P□ указывается непосредственно в виде адреса (указатель/метка).

Если в области смещения указаны два идентичных типа операндов, то значение смещения относится к операнду, расположенному последним в цепи контактов.

Команды IXDEV и IXSET необходимо применять совместно.

Величина смещения операнда ZR□.XX может находиться в диапазоне между 0 и 32767. Величина смещения является остатком от деления адреса операнда на 32767. Эта величина записывается в соответствующий регистр.

В качестве фиктивных контактов в области смещения допускаются только команды LD и AND. Все прочие команды игнорируются.

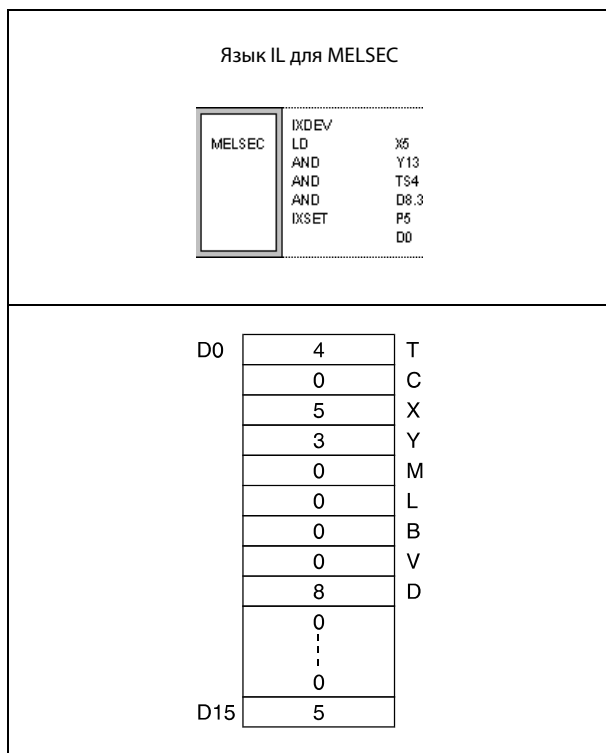


**Источники ошибок** В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Команды IXDEV и IXSET не запрограммированы в виде пары (код ошибки 4231).

**Пример** IXDEV, IXSET

Следующая программа записывает адреса (смещение) фиктивных контактов области смещения в соответствующий регистр. Величина смещения указателя/метки указана в команде IXSET. Взаимосвязь между типами операндов и регистрами указана в разделе, посвященном командам IX и IXEND.



## 7.7 Команды для работы с перечнями данных

Эти команды записывают данные в перечни и считывают данные из перечней. Текущие данные записываются в перечень и для дальнейшей обработки снова считываются в различной последовательности. Кроме того, эти команды позволяют стирать определенные блоки данных в перечне и вставлять в перечень новые блоки данных.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту МЭК (IEC)
Запись данных в перечень	FIFW	FIFW_M
	FIFWP	FIFWP_M
Считывание из перечня данных, введенных первыми	FIFR	FIFR_M
	FIFRP	FIFRP_M
Считывание из перечня данных, введенных последними	FPOP	FPOP_M
	FPOPP	FPOPP_M
Стирание определенных блоков данных в перечне	FDEL	FDEL_M
	FDELP	FDELP_M
Вставление определенных блоков данных в перечень	FINS	FINS_M
	FINSP	FINSP_M

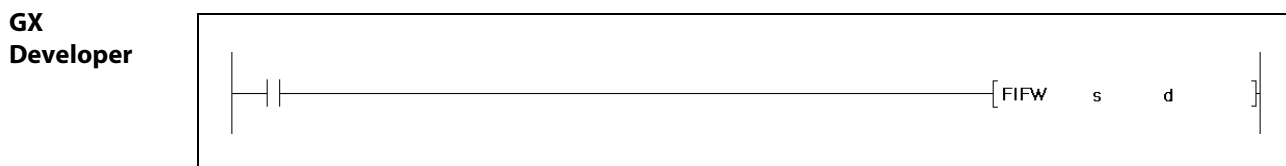
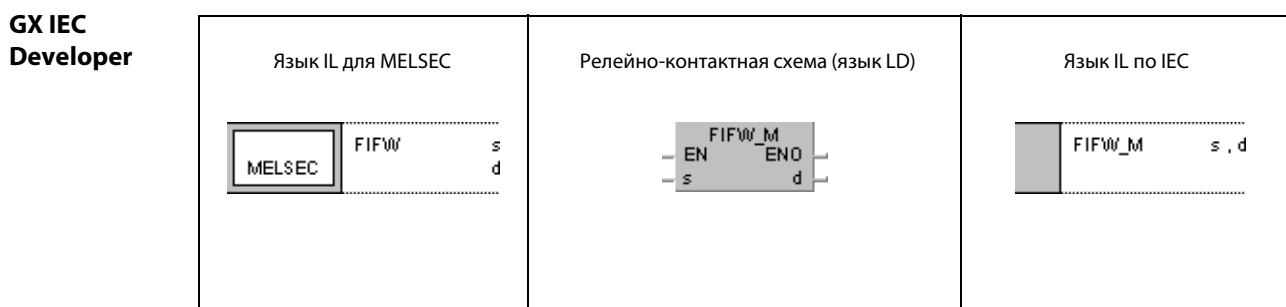
### 7.7.1 FIFW, FIFWP

<b>Процессор</b>	AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
	●	●	●	●	●	●

<b>Операнды MELSEC A</b>	Операнды																Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки					
	битовые						словные (16 бит)						Константы		Указатели							Уровень				
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P	I	N				M9012	M9010 M9011
	s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●				K1 ↓ K4	7	●		●
d							●	●	●	●																

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

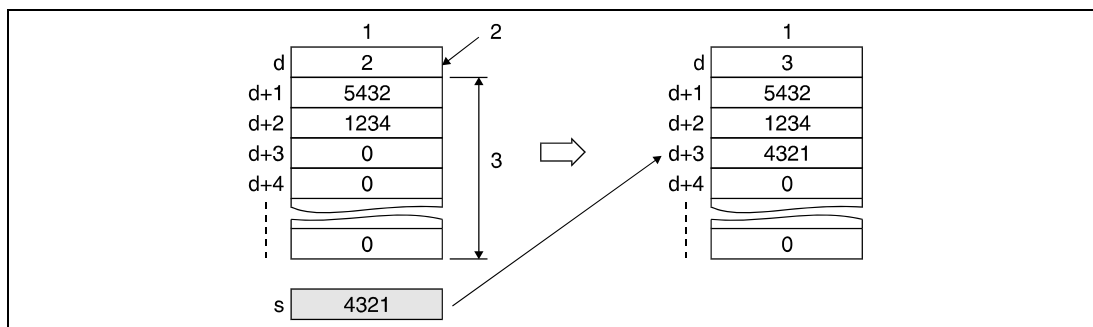
<b>Операнды MELSEC Q</b>	Операнды								Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□G□	Индексные регистры Zn	Константы		
	битовые	словные		битовые	словные					
	s	●	●	●	●	●	●	—	—	SMO
d	—	●	●	—	—	—	—	—		



<b>Переменные</b>	<b>Операнд</b>	<b>Значение</b>	<b>Тип данных</b>
	s	Записываемые в перечень данные или операнд, в котором сохранены эти данные.	BIN, 16 бит
	d	Первый адрес перечня данных.	

**Принцип действия****Запись данных в перечень****FIFW Команда записи данных**

Команда FIFW записывает в перечень определенную череду данных, указанную в *s*. Перечень задается областью адресов в *d* и располагает данные в последовательности их записи. В первом адресе области данных в *d* сохраняется количество блоков данных, содержащихся в перечне. Таким образом, значение в этом адресе является указателем для позиции новых данных, включаемых в перечень. При каждом выполнении команды FIFW это значение повышается на 1. Данные записываются начиная с адреса *d + 1*.



<sup>1</sup> перечень данных

<sup>2</sup> указатель позиции

<sup>3</sup> область адресов перечня данных

Перед первым выполнением команды FIFW необходимо стирать содержимое операнда, указанного в *d*.

При программировании пользователь должен учитывать количество записываемых блоков данных и требуемый для них диапазон адресов перечня.

Для администрирования различных блоков данных в различных перечнях следует использовать пользовательскую программу.

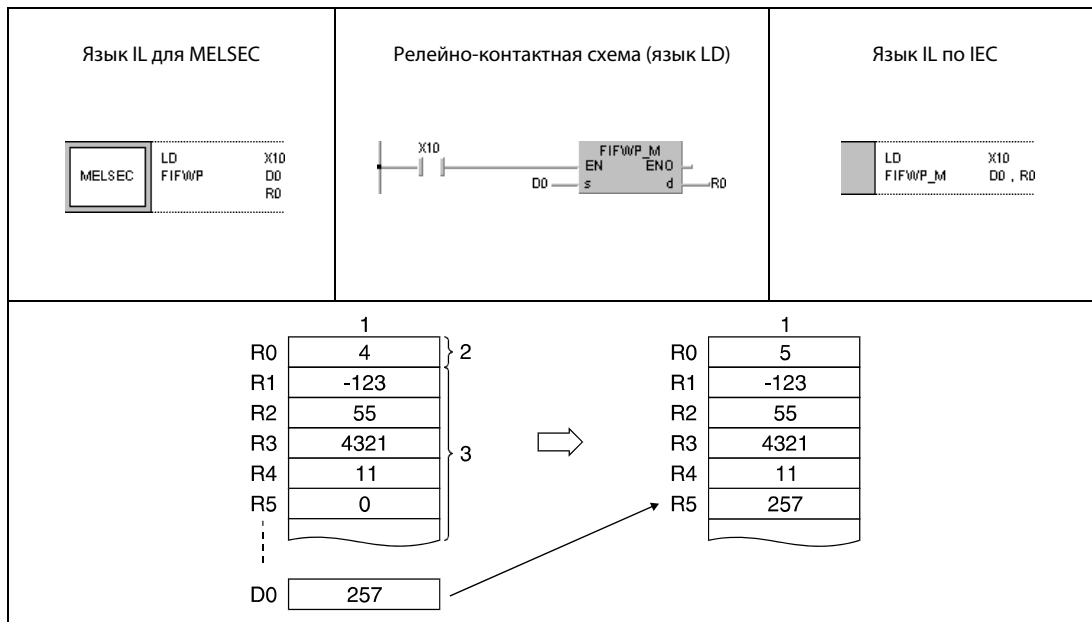
**Источники ошибок**

В следующем случае возникает ошибка обработки и устанавливается флаг ошибки:

- При выполнении команды FIFW длина перечня превышает предусмотренный для сохранения диапазон операнда (серии "Q"/"System Q" = код ошибки 4101).

**Пример 1** FIFWP

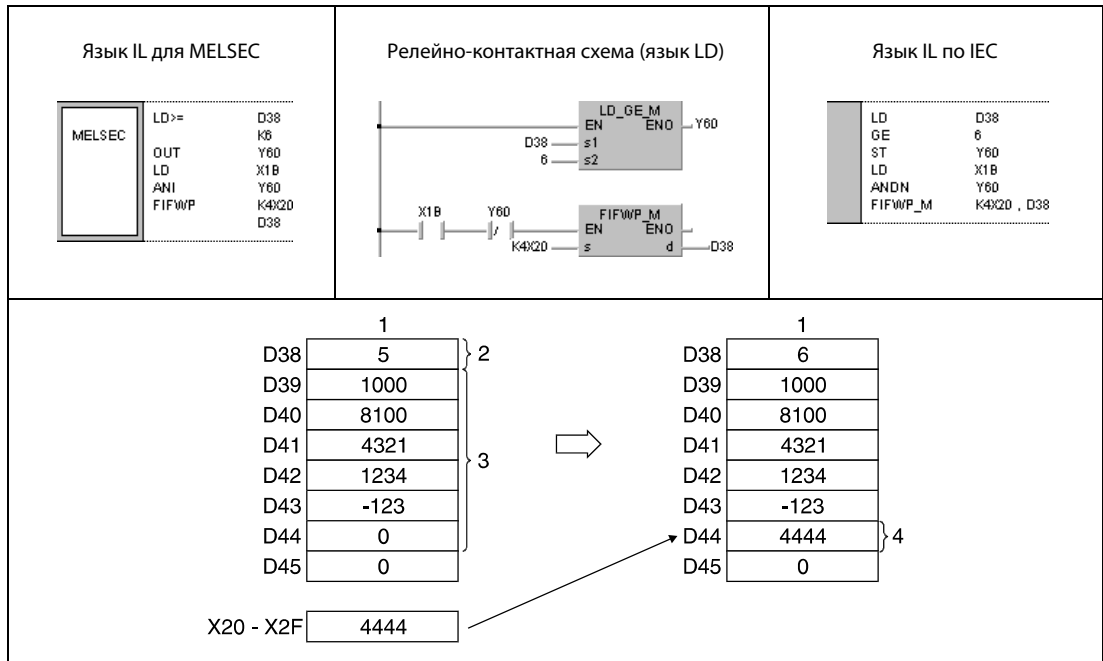
В следующей программе область памяти для перечня данных устанавливается с помощью регистров R0...R5. Начальный адрес (R0) этой области содержит информацию о количестве сохраненных блоков данных. При положительном фронте сигнала X10 данные из D0 записываются в ближайшую свободную позицию перечня данных (в примере: R5).



- <sup>1</sup> перечень данных
- <sup>2</sup> указатель позиции
- <sup>3</sup> область адресов перечня данных

**Пример 2** FIFWP

В следующей программе область памяти для перечня данных задается с помощью регистров данных D38...D44. Начальный адрес (D38) этой области содержит информацию о количестве сохраненных блоков данных. При положительном фронте X1B данные входов X20...X2F записываются в ближайшую свободную позицию перечня данных (в примере: D44). В заданном здесь перечне можно сохранить максимум 6 блоков данных. Поэтому предусмотрена блокировка команды FIFW с помощью выхода Y60. Этот выход включается, если содержимое регистра D38 больше или равно 6.



- 1 перечень данных
- 2 указатель позиции
- 3 область адресов перечня данных
- 4 последний адрес сохранения

### 7.7.2 FIFR, FIFRP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

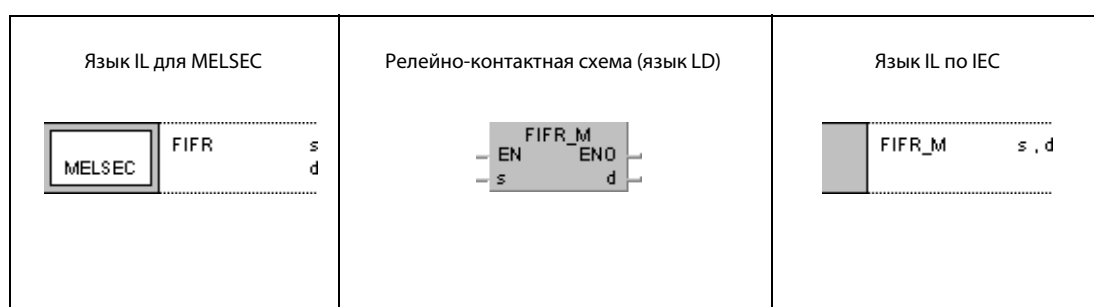
	Операнды															Длина блока K1 ↓ K4	Шагов 7 ↓ 1	Индекс ●	Флаг пере- носа M9012	Флаг ошибки M9010 M9011
	битовые					словные (16 бит)					Константы		Указа- тели		Уро- вень					
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z					
s	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●					
d						●	●	●	●	●										

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

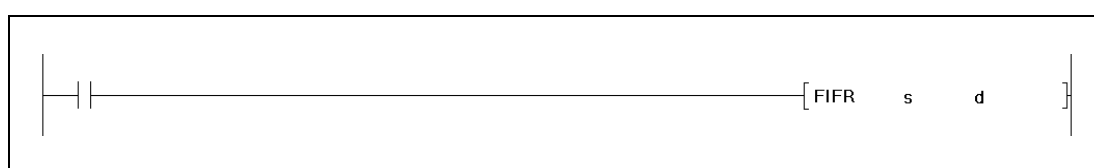
**Операнды MELSEC Q**

	Операнды								Флаг ошибки SMO	Шагов 3	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□□G□	Индексные регистры Zn	Константы			Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	—	—		
d	—	●	●	—	—	—	—	—	—		

**GX IEC Developer**



**GX Developer**



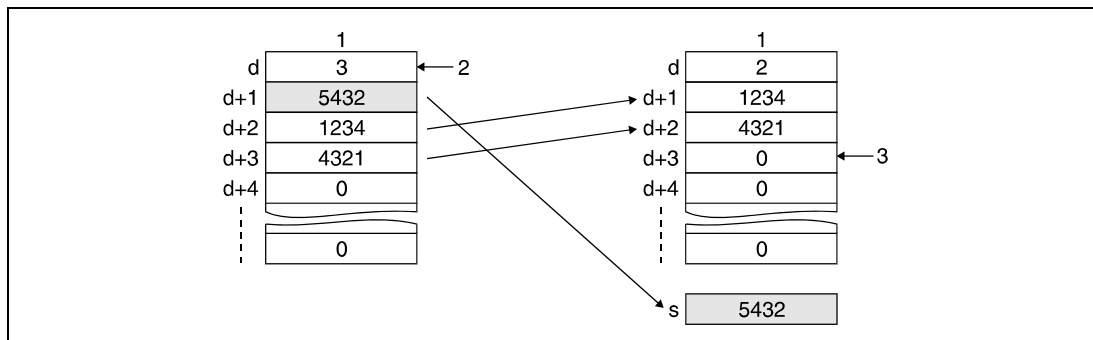
**Переменные**

Операнд	Значение	Тип данных
s	Начальный адрес операнда, в котором сохраняются считанные данные.	BIN, 16 бит
d	Начальный адрес перечня	

**Принцип действия****Считывание из перечня данных, введенных первыми****FIFR Команда считывания данных, введенных первыми**

Команда FIFR считывает данные из перечня и сохраняет их в заданной области адресов. Считывание данных начинается с первого адреса  $d + 1$  после указателя позиции. Данные передаются в область адресов, указанную в  $s$ .

Данные перечня сдвигаются к началу перечня в последовательности их записи в перечень. Все предыдущие данные стираются. После считывания позиция указателя (первый адрес в  $d$ ) понижается на 1.



<sup>1</sup> перечень данных

<sup>2</sup> указатель позиции

<sup>3</sup> В этот регистр записывается значение 0.

**ПРИМЕЧАНИЕ**

Учитывайте, что если значение в  $d$  (указатель позиции) равно 0, команда не вызывается.

**Источники ошибок**

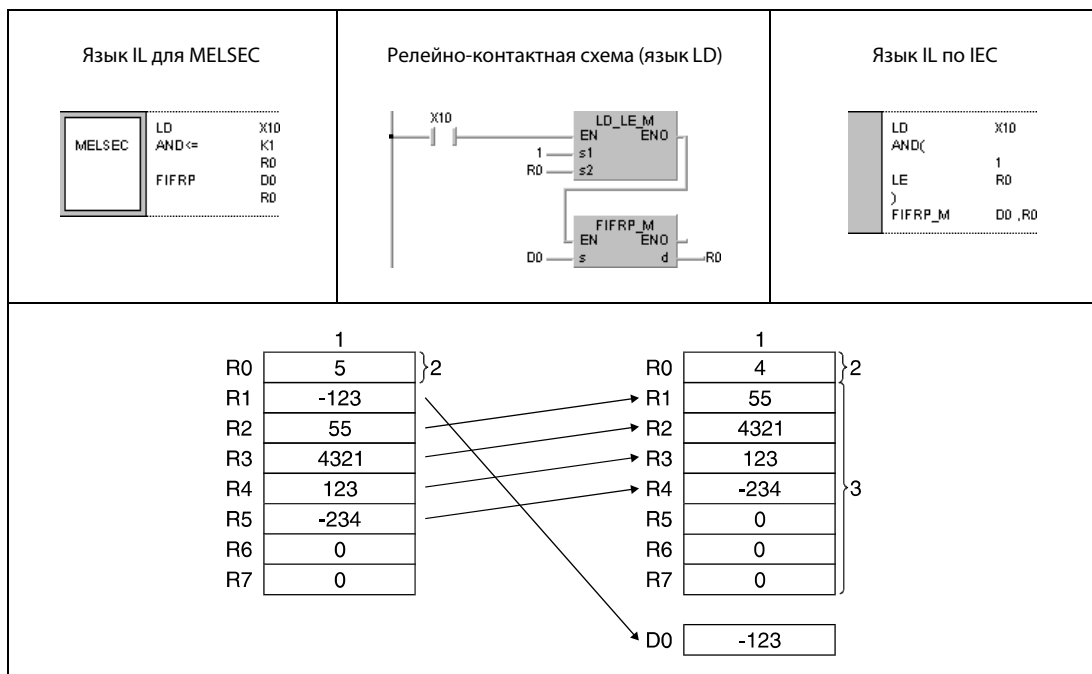
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Команда FIFR выполняется при указателе позиции, равном 0 (серии "Q"/"System Q" = код ошибки 4100).
- При выполнении команды FIFR длина перечня превышает область операндов (серии "Q"/"System Q" = код ошибки 4101).



**Пример 1** FIFRP

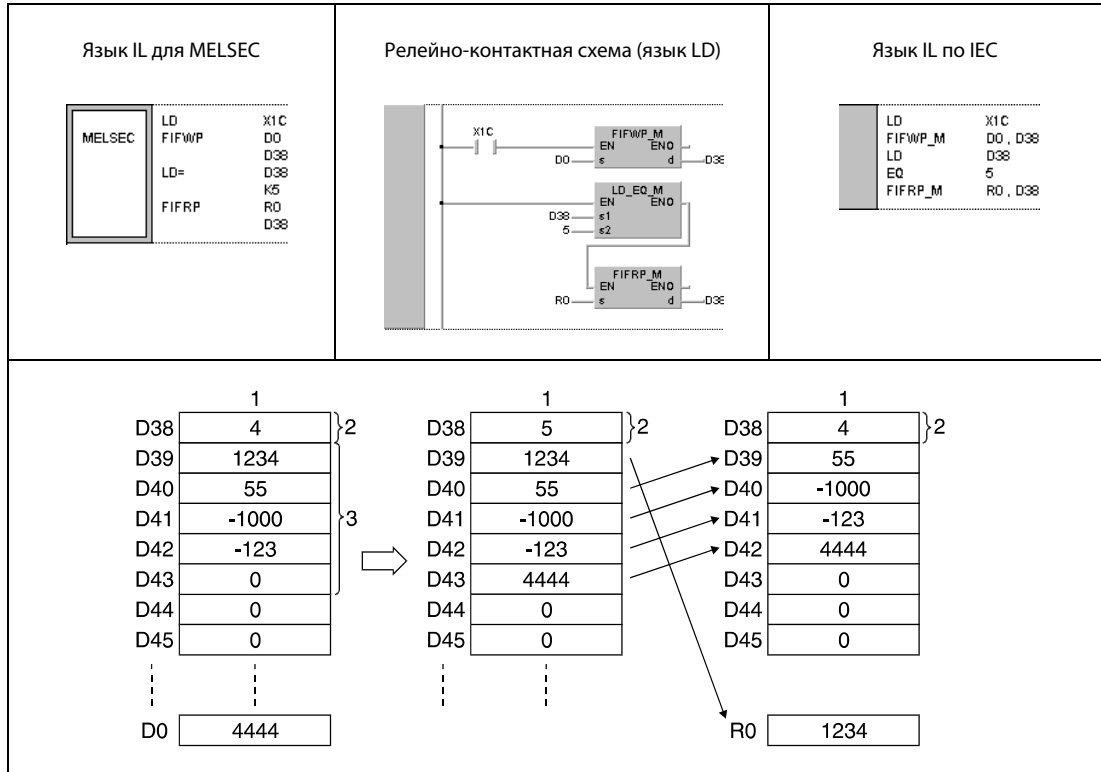
В следующей программе при положительном фронте сигнала X10 из перечня данных R0...R7 считывается значение R1 (значение, введенное первым) и это значение сохраняется в регистре D0. В примере указатель позиции в начале выполнения имеет значение 5, а после выполнения 4. Предшествующая команда сравнения предотвращает выполнение команды FIFR, если указатель позиции (R0) содержит значение 0.



- <sup>1</sup> перечень данных
- <sup>2</sup> указатель позиции
- <sup>3</sup> область адресов перечня данных

**Пример 2** FIFRP

Следующая программа при каждом положительном фронте X1C записывает значение из D0 в перечень данных D38...D43. Если значение указателя позиции равно 5, первое значение перечня, действующего по принципу FIFO (т. е. "первым пришел – первым ушел"), считывается и передается на R0. Этот процесс повторяется при каждом положительном фронте X1C.



- <sup>1</sup> перечень данных
- <sup>2</sup> указатель позиции
- <sup>3</sup> область адресов перечня данных

### 7.7.3 FPOP, FPOPP

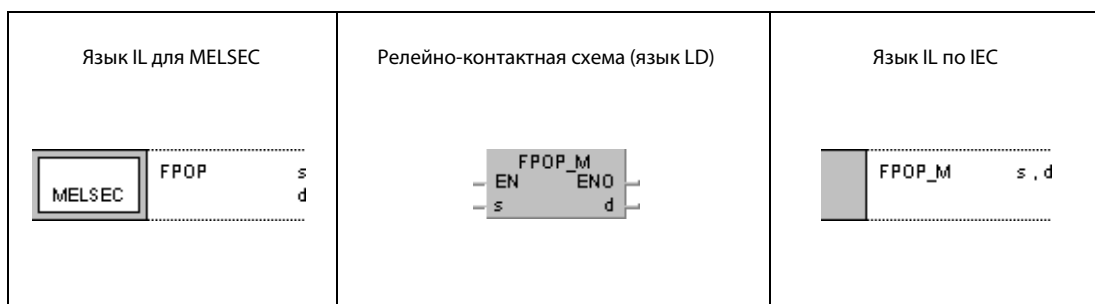
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

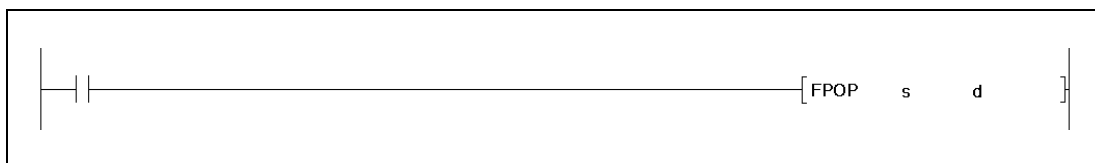
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы			Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	●	—	SMO	3
d	—	●	●	—	—	—	—	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Начальный адрес операнда, в котором сохраняются считанные данные.	BIN, 16 бит
d	Начальный адрес перечня данных	

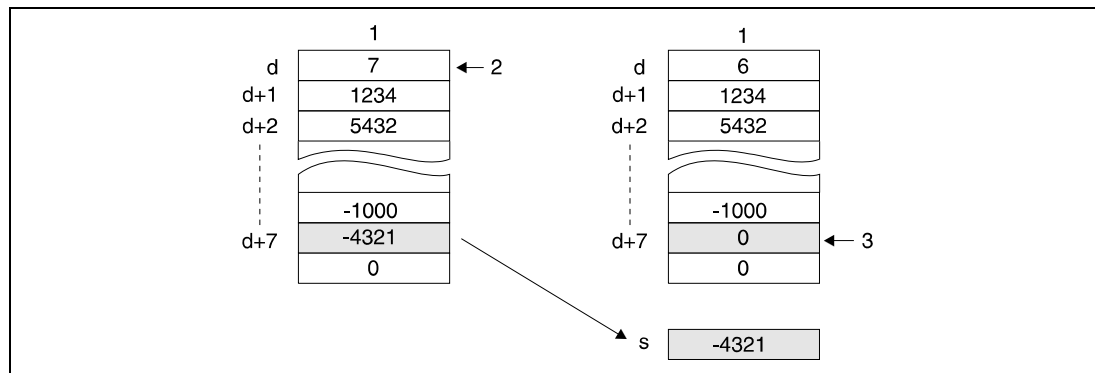
**Принцип действия**

**Считывание из перечня данных, введенных последними**

**FPOP Команда для считывания данных, введенных последними**

Команда FPOP считывает данные из перечня и сохраняет их в заданной области адресов. Считывание начинается с адреса  $d + n$  перечня, записанного последним. Данные передаются в область адресов, указанную в  $s$ .

В уже считанный адрес перечня записывается значение 0. После считывания позиция указателя (первый адрес в  $d$ ) понижается на 1.



- <sup>1</sup> перечень данных
- <sup>2</sup> указатель позиции
- <sup>3</sup> В этот регистр записывается значение 0.

**ПРИМЕЧАНИЕ** Учитывайте, что если значение в  $d$  (указатель позиции) равно 0, команда не вызывается.

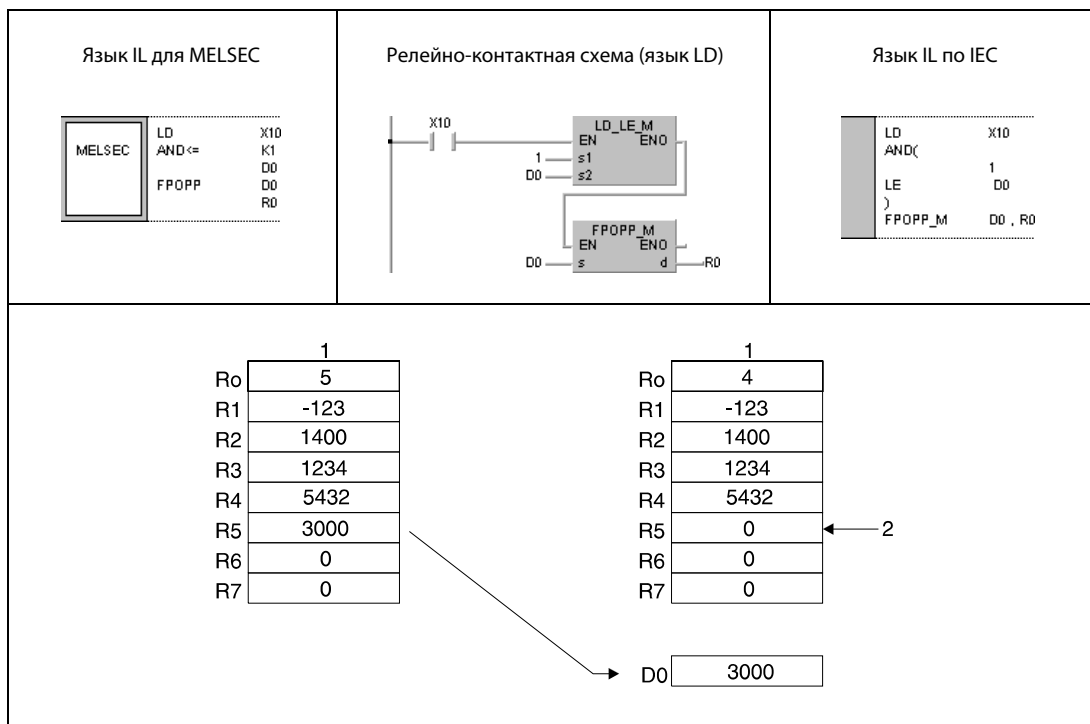
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Команда FPOP выполняется при указателе позиции, равном 0 (код ошибки 4100).
- При выполнении команды FPOP длина перечня данных превышает область операндов (код ошибки 4101).

**Пример 1** FPOP

В следующей программе при положительном фронте сигнала X10 из перечня R0...R7 считывается значение адреса R5 (значение, введенное последним) и это значение сохраняется в регистре D0. В примере указатель позиции в начале выполнения имеет значение 5, а после выполнения 4. Предшествующая команда сравнения предотвращает выполнение команды FPOP, если указатель позиции (R0) имеет значение 0.

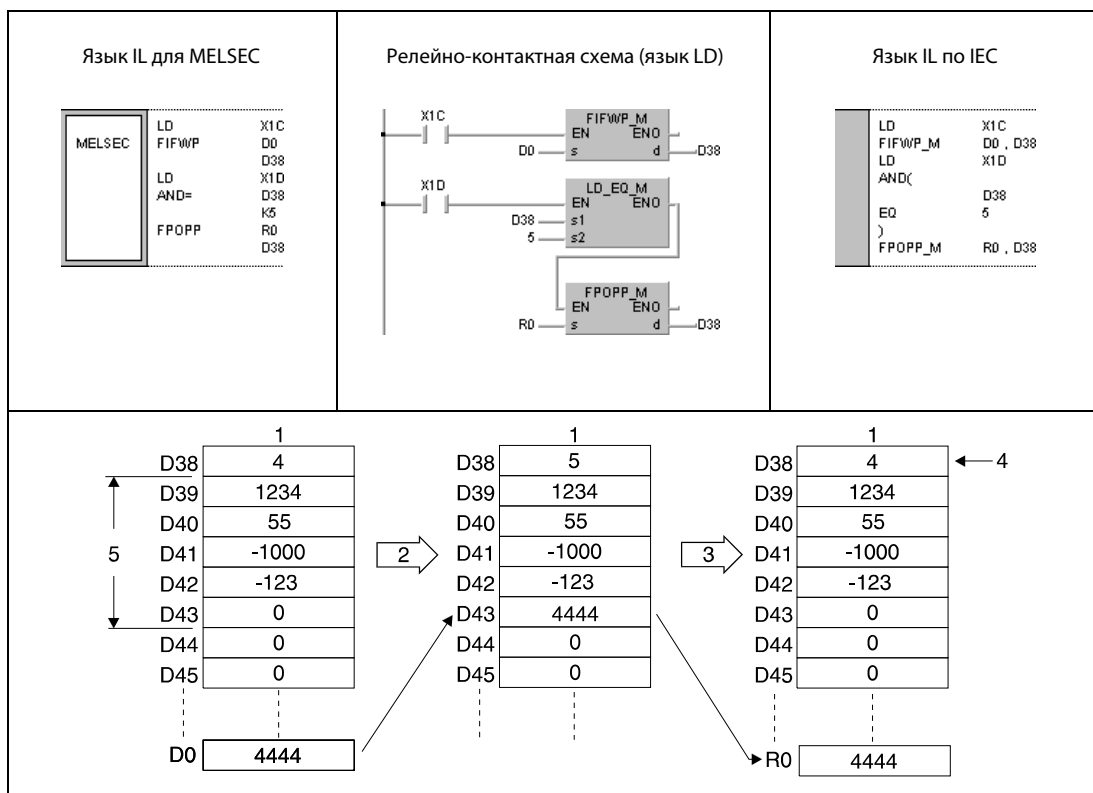


<sup>1</sup> перечень данных

<sup>2</sup> В этот регистр записывается значение 0.

**Пример 2** FROPP

Следующая программа при каждом положительном фронте X1C записывает значение из D0 в перечень данных D38...D43. Если указатель позиции имеет значение 5, то при положительном фронте сигнала X1D значение регистра D43 считывается и передается в R0.



- 1 перечень данных
- 2 положительный фронт X1C
- 3 положительный фронт X1D
- 4 указатель позиции
- 5 текущий диапазон адресов перечня

### 7.7.4 FDEL, FDELP, FINS, FINSP

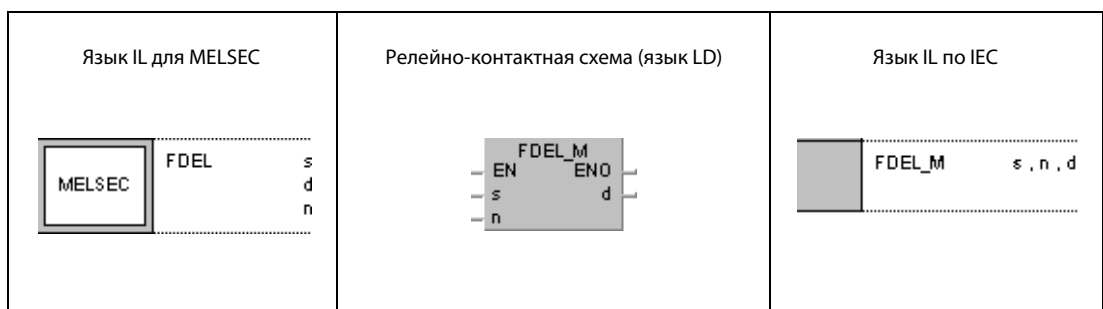
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

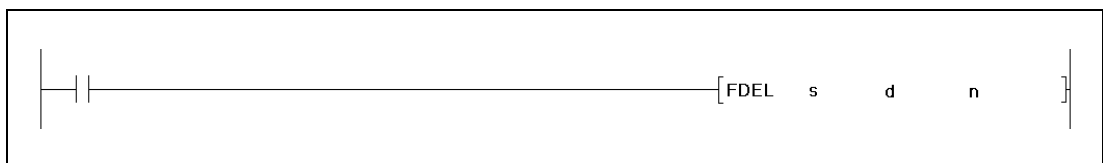
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	—	—	SM0	4
d	—	●	●	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Данные, вставляемые в перечень по определенному адресу, или операнд, в котором сохранены эти данные. Первый адрес операнда, в котором сохраняются данные, стираемые в перечне по определенному адресу.	BIN, 16 бит
d	Начальный адрес перечня данных	
n	Адреса, по которому вводятся или стираются данные.	

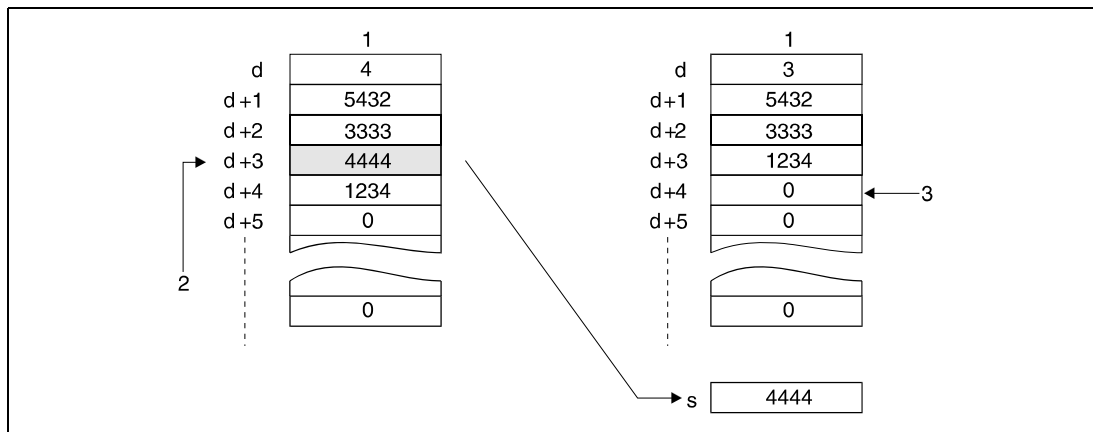
**Принцип действия**

**Стирание определенных блоков данных в перечне данных и вставка определенных блоков данных в перечень данных**

**FDEL Стирание определенных блоков данных**

Команда FDEL стирает n-ый блок данных после указателя позиции в перечне данных, указанном в d, и сохраняет стертое значение в операнде, указанном в s.

После стирания блока данные перечня смыкаются. После считывания позиция указателя (первый адрес в d) понижается на 1.



<sup>1</sup> перечень данных

<sup>2</sup> При n = 3 стирается блок данных d + 3.

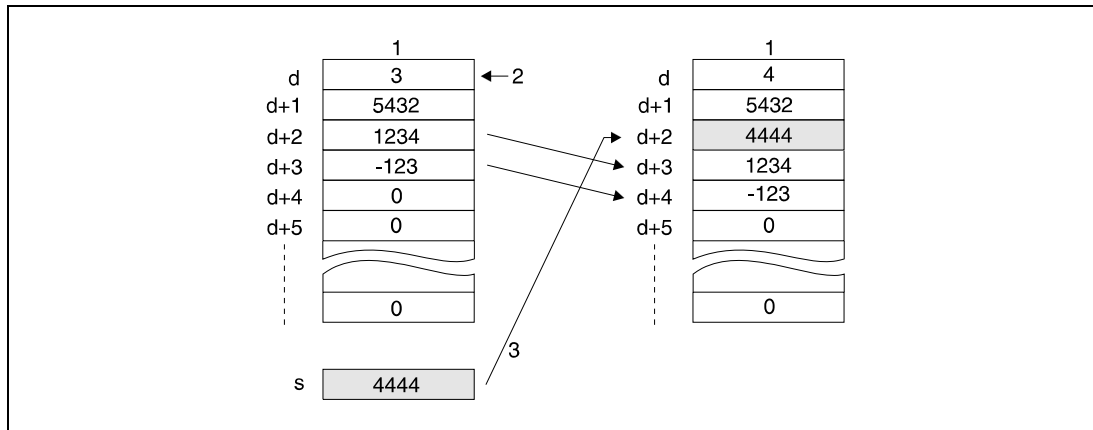
<sup>3</sup> В этот регистр записывается значение 0.



**FINS Вставление определенных блоков данных**

Команда FINS вставляет указанный в *s* 16-битный блок данных в *n*-ную позицию после указателя позиции в перечне данных, указанном в *d*.

Блоки данных, следующие за местом вставления, сдвигаются на один адрес дальше. После вставления значение указателя позиции (первый адрес в *d*) повышается на 1.



<sup>1</sup> перечень данных

<sup>2</sup> указатель позиции

<sup>3</sup> При  $n = 2$  блок данных вставляется в  $d + 2$ .

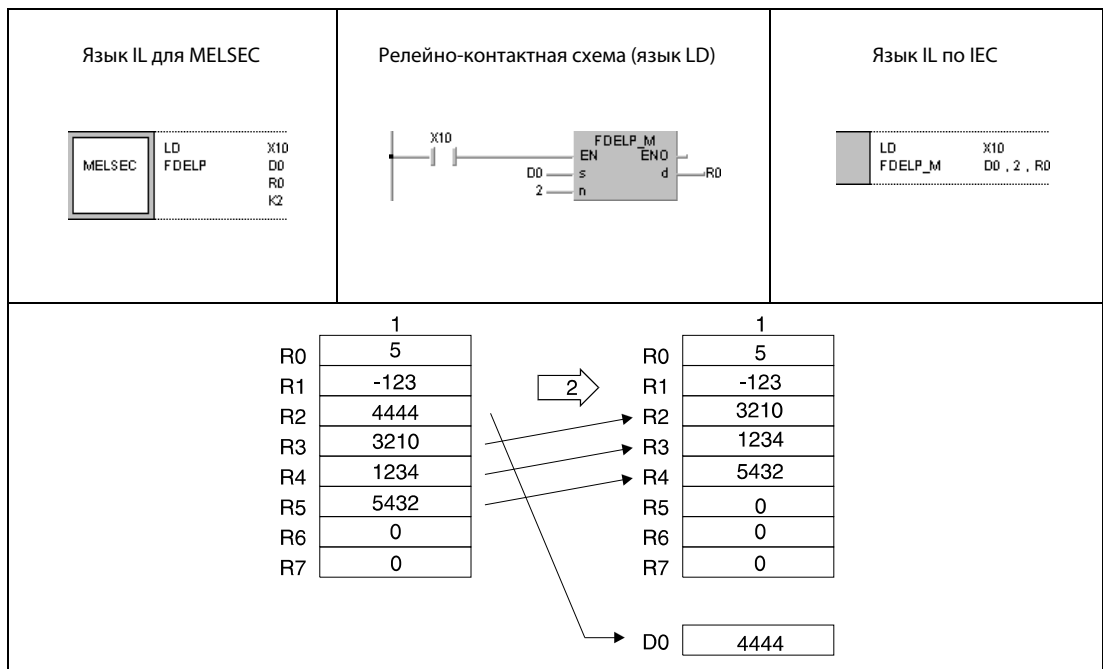
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- При выполнении команды FDEL или FINS указанный в *n* операнд превышает область операнда, указанного в *d* (код ошибки 4101).
- При выполнении команды FDEL или FINS операнд в *d*, указанный с помощью *n*, находится вне диапазона адресов, образуемого уже имеющимися блоками данных + 1 (код ошибки 4101).
- Команда FDEL или FINS выполнена при  $n = 0$  (код ошибки 4100).
- Команда FDEL выполнена при значении *d*, равном 0 (код ошибки 4100).
- При выполнении команды FPOP или FINS длина перечня данных превышает область операндов (код ошибки 4101).

**Пример 1** FDELP

Следующая программа при положительном фронте сигнала X10 стирает второй блок данных (R2) в перечне R0...R7 и сохраняет стертое значение в D0.

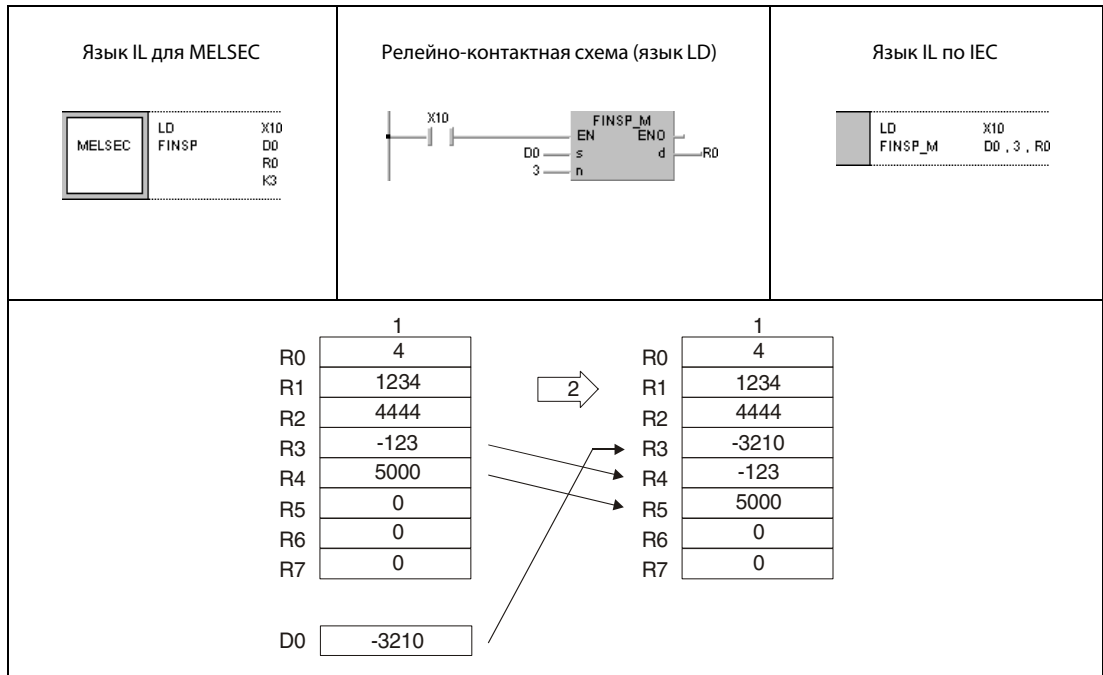


<sup>1</sup> перечень данных

<sup>2</sup> положительный фронт X10

**Пример 2** FINSP

Следующая программа при положительном фронте сигнала X10 вставляет блок данных, указанный в D0, в качестве блока данных R3 перечня R0...R7.



<sup>1</sup> перечень данных

<sup>2</sup> положительный фронт X10



## 7.8 Команды доступа к буферной памяти

Нижеописанные команды обращаются к буферной памяти специальных модулей. Эти команды позволяют центральному процессору обмениваться данными с соответствующими модулями. Обзор команд дан в следующей таблице.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту МЭК (IEC)
Считывание данных из специального модуля	FROM	FROM_M
	FROMP	FROMP_M
	DFRO	DFRO_M
	DFROP	DFROP_M
Запись данных в специальный модуль	TO	TO_M
	TOP	TOP_M
	DTO	DTO_M
	DTOP	DTOP_M

7.8.1 FROM, FROMP, DFRO, DFROP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

	Операнды															Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки						
	битовые					словные (16 бит)					Константы		Указатели		Уровень											
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V	K	H (16#)	P	I	N
n1																	●	●				K1 ↓ K4 2 ↓ K1 K8	9 3	●		●
n2																										
d	●	●	●	●	●	●	●	●	●	●	●						●	●								
n3																	●	●								

<sup>1</sup> В процессоре A3H обработка битовых операндов не возможна.

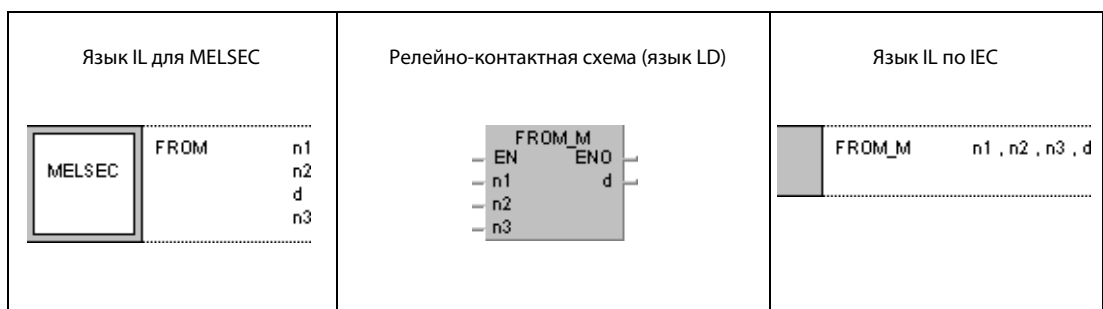
<sup>2</sup> При программировании команды FROM/P длина блока может быть в диапазоне между K1 и K4, а при программировании команды DFRO/P – между K1 и K8.

<sup>3</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

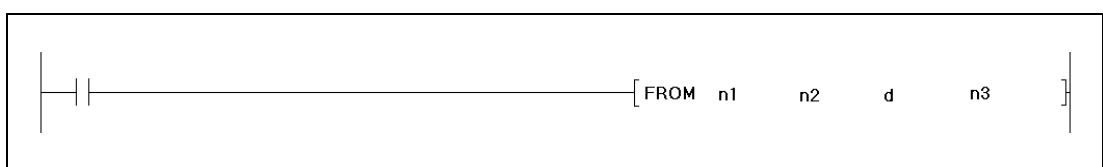
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные U
	битовые	словные		битовые	словные						
n1	●	●	●	●	●	●	●	●	●	SM0	5
n2	●	●	●	●	●	●	●	●			
d	●	●	●	—	—	—	—	—			
n3	●	●	●	●	●	●	●	●			

GX IEC Developer



GX Developer

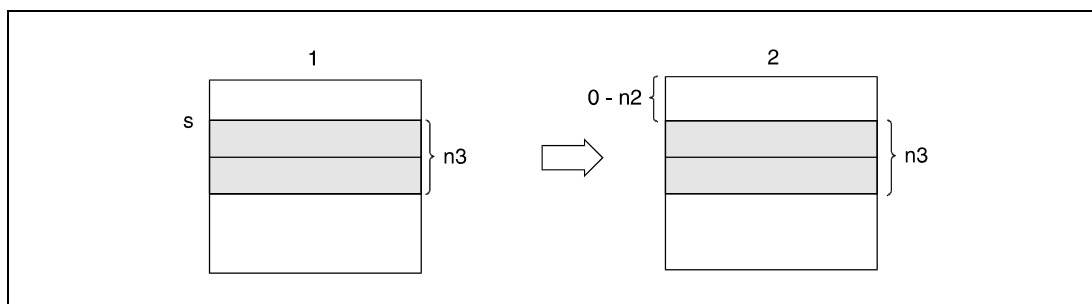


Переменные

Операнд	Значение	Тип данных
n1	Головной адрес специального модуля на монтажной шине.	BIN, 16 бит
n2	Начальный адрес области, из которой происходит считывание.	BIN, 16/32 бита
d	Начальный адрес области центрального процессора, в которую производится запись.	BIN, 16 бит
n3	Количество считываемых слов данных.	

**Принцип действия****Считывание 1-словных и 2-словных данных (32 бита) из специального модуля****FROM Считывание 1-словных данных (16 бит)**

Команда FROM считывает однословные данные из области буферной памяти специального модуля и сохраняет их в заданной области адресов центрального процессора. Начальный адрес считываемых данных задается в  $n2$ , количество слов данных – в  $n3$ , а адрес специального модуля, связанный с местоположением модуля на монтажной шине – в  $n1$ . Диапазон адресов центрального процессора, в котором должны сохраняться данные, указывается в  $d$ .



<sup>1</sup> буферная память специального модуля

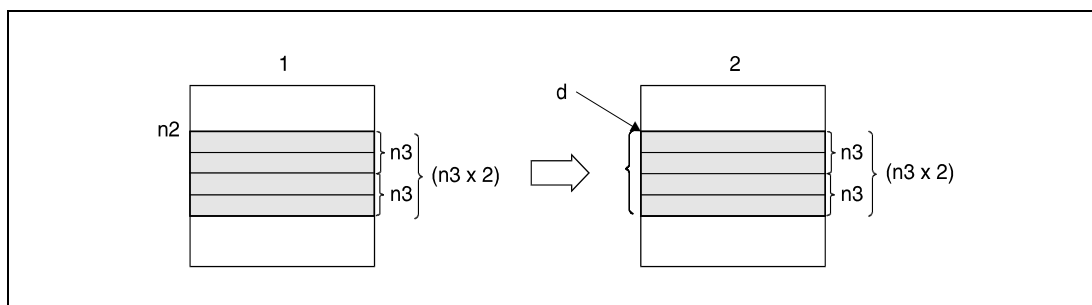
<sup>2</sup> память центрального процессора

**ПРИМЕЧАНИЕ**

В многопроцессорном режиме с помощью команды FROM можно также считывать данные из другого центрального процессора. Более подробная информация на эту тему имеется в разделе 9.6.2.

**DFRO Считывание 2-словных данных (32 бита)**

Команда DFRO считывает двухсловные данные из буферной памяти специального модуля. Начальный адрес считываемых данных задается в  $n2$ , количество слов данных (четное) – в  $n3$ , а адрес специального модуля – в  $n1$ . Диапазон адресов центрального процессора, в котором должны сохраняться данные, указывается в  $d$ .



<sup>1</sup> буферная память специального модуля

<sup>2</sup> память центрального процессора

**ПРИМЕЧАНИЕ**

Кроме того, центральные процессоры серий "Q" и "System Q" могут обращаться к буферной памяти специальных модулей непосредственно. В этом случае операнды указываются в виде  $U \square \backslash G \square$  ( $U$  (головной адрес специального модуля)/ $G$  (начальный адрес в буферной памяти)).

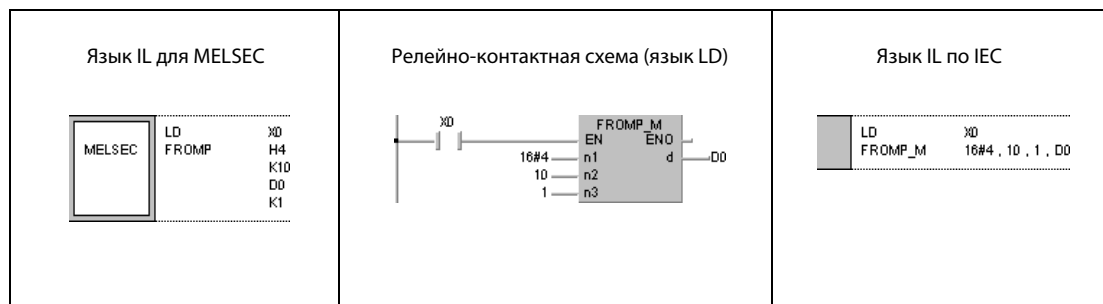
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Перед выполнением команды не было обмена данными со специальным модулем (серия "Q"/"System Q" = код ошибки 1412).
- Перед выполнением команды возникла неполадка в специальном модуле (серия "Q"/"System Q" = код ошибки 1402).
- По указанному в n1 адресу ввода-вывода не имеется специального модуля (серия "Q"/"System Q" = код ошибки 2110).
- Заданное в n3 количество слов данных превышает предусмотренную для сохранения область операнда в d (серия "Q"/"System Q" = код ошибки 4101).
- Указанный в n2 адрес находится вне диапазона буферной памяти (серия "Q"/"System Q" = код ошибки 4100).
- В n2 указан неправильный адрес (AJ71QC24) (серия "Q"/"System Q" = код ошибки 4100).
- Доступ к специальному модулю не возможен.

**Пример 1****FROMP**

В следующей программе при положительном фронте сигнала X0 считываются цифровые значения канала CH1 из адреса 10 буферной памяти модуля A68AD. Диапазон адресов модуля: от 040 до 05F. Считанные данные сохраняются в D0.

**Пример 2****DFROP**

В следующей программе при положительном фронте сигнала X0 считываются данные оси "X" с адресов 602 и 603 буферной памяти модуля AD71. Диапазон адресов модуля: от 040 до 05F. Считанные данные сохраняются в D0 и D1.

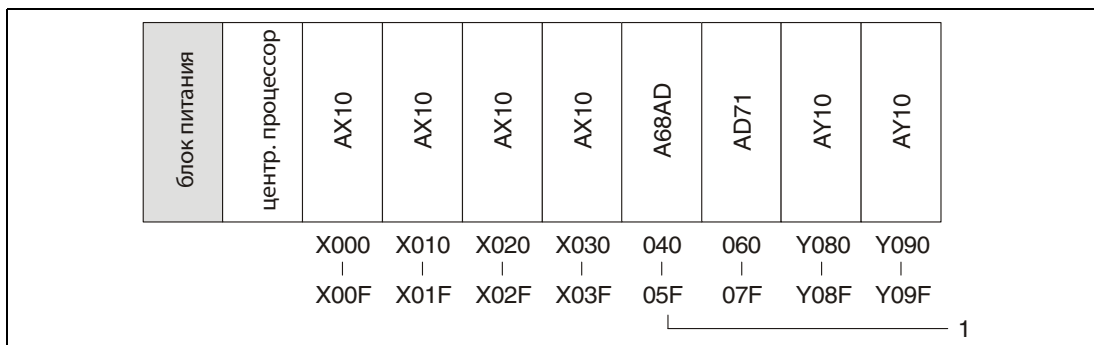




**ПРИМЕЧАНИЯ** Указываемый в *n1* головной адрес устанавливается следующим образом:

*n1* = 10 ⇒ головной адрес = 1

*n1* = 20 ⇒ головной адрес = 2



<sup>1</sup> начальный адрес специального регистра: *n1* = K4 или H4

Без объявления переменных в заголовке *POU* пример 2 программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

7.8.2 TO, TOP, DTO, DTOP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

Операнды MELSEC A

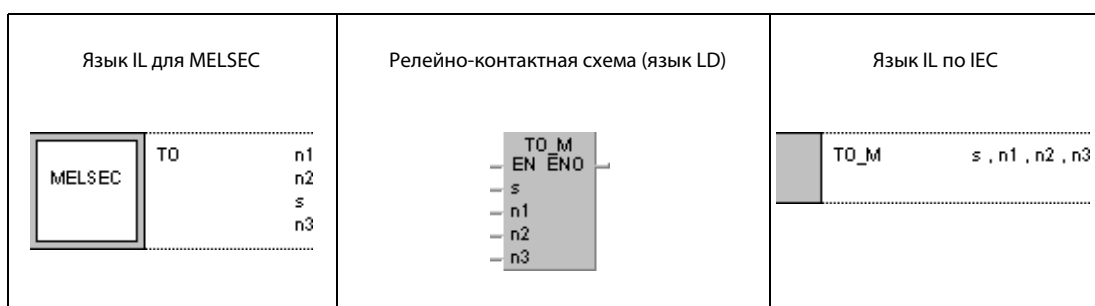
	Операнды																Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки	
	битовые						словные (16 бит)						Константы		Указатели							Уровень
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						
n1																	●	●				
n2																	●	●				
s	● <sup>1</sup>	● <sup>1</sup>	● <sup>1</sup>	● <sup>1</sup>	● <sup>1</sup>	● <sup>1</sup>	● <sup>1</sup>	● <sup>1</sup>	● <sup>1</sup>	● <sup>1</sup>	● <sup>1</sup>	● <sup>1</sup>					● <sup>2</sup>	● <sup>2</sup>				
n3																	●	●				

- <sup>1</sup> В процессоре A3N обработка битовых операндов не возможна.
- <sup>2</sup> Возможный диапазон констант для s: от H0 до FFFF, от K-32768 до 32767
- <sup>3</sup> При программировании команды TOP длина блока может быть в диапазоне от K1 до K4, а при программировании команды DTOP – от K1 до K8.
- <sup>4</sup> Количество шагов при использовании процессоров AnA или AnAS указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

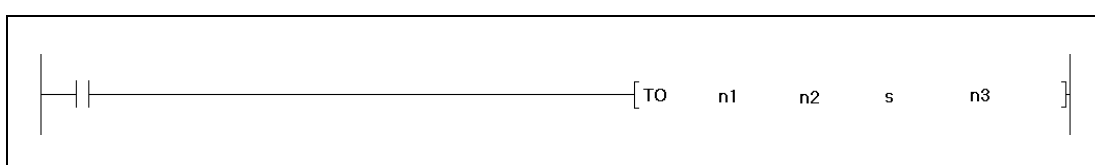
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непос. адр. J□□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные U
	битовые	словные		битовые	словные						
n1	●	●	●	●	●	●	●	●	●	SM0	5
n2	●	●	●	●	●	●	●	●	—		
s	●	●	●	—	—	—	—	●	—		
n3	●	●	●	●	●	●	●	●	—		

GX IEC Developer



GX Developer



**Переменные**

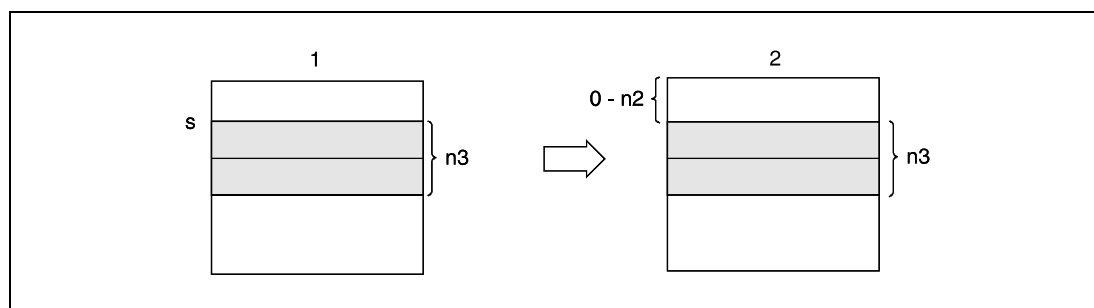
Операнд	Значение	Тип данных
n1	Головной адрес специального модуля на монтажной шине.	BIN, 16 бит
n2	Начальный адрес области, в которую производится запись.	
s	Записываемые данные или начальный адрес области центрального процессора, в которой сохранены записываемые данные.	BIN, 16/32 бита
n3	Количество записываемых слов данных.	BIN, 16 бит

**Принцип действия**

**Запись в буферную память**

**TO Запись 1-словных данных**

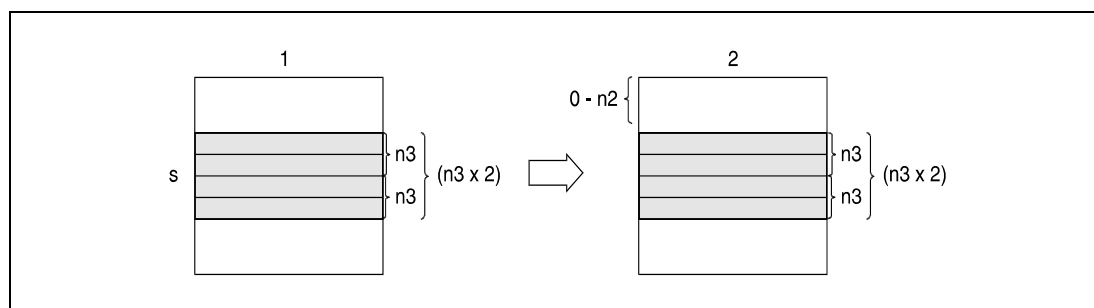
Команда TO записывает 1-словные данные из памяти центрального процессора в буферную память специального модуля. Начальный адрес области памяти, в которую требуется передать данные, задается в n2, количество слов данных – в n3, а адрес специального модуля, связанный с местоположением модуля на монтажной шине – в n1. Начальный адрес диапазона адресов, из которого требуется считать данные, устанавливается в s.



- <sup>1</sup> память центрального процессора
- <sup>2</sup> буферная память специального модуля

**DTO Запись 2-словных данных (32 бита)**

Команда DTO записывает 2-словные данные (32 бита) в буферную память специального модуля. Начальный адрес области памяти, в которую требуется передать данные, задается в n2, количество слов данных (четное) – в n3, а адрес специального модуля – в n1. Начальный адрес диапазона адресов, из которого требуется считать данные, устанавливается в s.



- <sup>1</sup> память центрального процессора
- <sup>2</sup> буферная память специального модуля

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Перед выполнением команды не было обмена данными со специальным модулем (серия "Q"/"System Q" = код ошибки 1412).
- Перед выполнением команды возникла неполадка в специальном модуле (серия "Q"/"System Q" = код ошибки 1402).
- По указанному в n1 адресу ввода-вывода не имеется специального модуля (серия "Q"/"System Q" = код ошибки 2110).
- Заданное в n3 количество слов данных превышает предусмотренную для сохранения область операнда в s (серия "Q"/"System Q" = код ошибки 4101).
- Указанный в n2 адрес находится вне диапазона буферной памяти (код ошибки 4100).
- В n2 указан неправильный адрес (AJ71QC24) (серия "Q"/"System Q" = код ошибки 4100).
- Доступ к специальному модулю не возможен.

**Пример 1**

**TOP**

В следующей программе при положительном фронте сигнала X0 включается аналогоцифровое преобразование каналов CH1 и CH2 модуля A68AD. Специальный модуль занимает адреса с 040 по 05F. По адресу 0 буферной памяти записывается значение 3.



**Пример 2**

**DTOP**

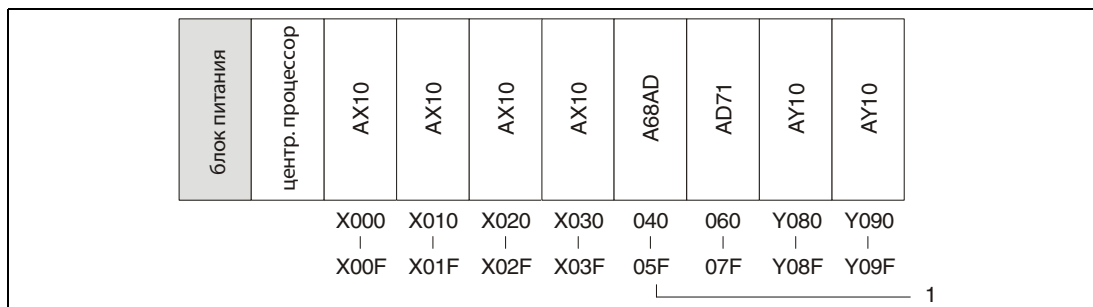
В следующей программе при положительном фронте сигнала X0 значения данных X на адресах буферной памяти var\_D0 модуля AD71 устанавливаются на 0. Специальный модуль расположен на адресах с 040 по 05F.



**ПРИМЕЧАНИЯ** Указываемый в n1 головной адрес устанавливается следующим образом:

n1 = 10 ⇒ головной адрес = 1

n1 = 20 ⇒ головной адрес = 2



<sup>1</sup> начальный адрес специального регистра: n1 = K4 или H4

Без объявления переменных в заголовке POU пример 2 программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

## 7.9 Команды индикации

Контроллеры MELSEC Q и MELSEC A имеют ряд команд, с помощью которых знаки ASCII можно выводить на выходы выходного модуля или отображать на светодиодном дисплее, имеющемся с передней стороны модулей центральных процессоров (имеющих соответствующее оснащение). В общей сложности имеются 7 различных команд индикации.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту МЭК (IEC)
Вывод ASCII	PR	PR_M
	PRC	PRC_M
Индикация знаков ASCII и комментариев	LED	LED_M
	LEDC	LEDC_M
	LEDA	LEDA_M
	LEDB	LEDB_M
Стирание индикации	LEDR	LEDR_M

**ПРИМЕЧАНИЯ** Для процессора A3A команды LEDA и LEDB не могут быть непосредственно использованы в качестве команд индикации. В этом процессоре эти команды служат для запуска специализированных прикладных команд.

Чтобы описываемую в этом разделе функцию команд LEDA и LEDB можно было использовать в процессоре A3A, строковые данные необходимо изменить с помощью специализированных прикладных команд для процессоров AnA или AnAS. Более подробная информация на эту тему имеется в отдельном руководстве по программированию процессоров серий AnA и AnAS (раздел "Dedicated Instructions" или "Специализированные команды").

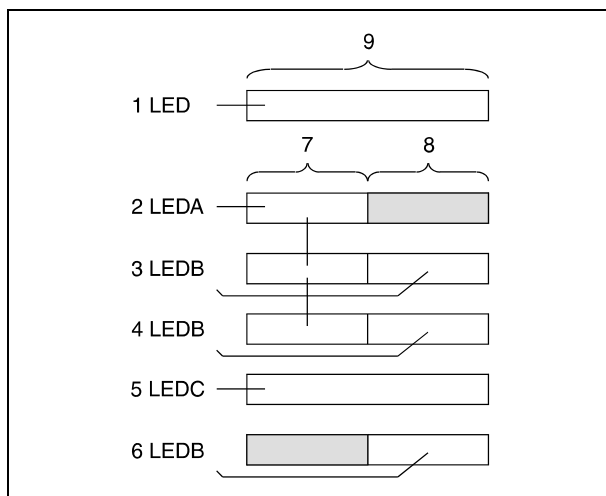
Для индикации на светодиодном дисплее центрального процессора установлены следующие приоритеты:

1. Индикация ошибок, выявленных при самодиагностике
2. Индикация команды CHK
3. Индикация номера маркера ошибки F
4. Индикация знаков ASCII по команде LED(A,B,C)
5. "BATTERY ERROR"

При использовании процессора A3A этот порядок приоритетов можно произвольно изменить. Более подробная информация имеется в руководствах по серии AnA.

Если дисплей отображает сообщение одного из трех первых приоритетов, то выполнение команды индикации не изменяет содержимое индикации сразу. Если на дисплее высвечено сообщение "BATTERY ERROR", то при выполнении команды LED(A,B,C) индикация изменяется сразу.

На рисунке схематически показана индикация на светодиодном дисплее после выполнения команды LED (A,B,C).



Команда LED отображает на дисплее (1) до 16 знаков (9). При выполнении команды LEDA используются первые 8 знаков (7). Последние 8 разрядов (8) остаются свободными (2). Если затем выполняется команда LEDB, то данные отображаются во второй половине дисплея (3). Повторное выполнение команды LEDB перезаписывает первоначальные данные во второй половине. Данные первых 8 знаков остаются без изменений (4). При выполнении команды LEDC на дисплей выводится предварительно заданный комментарий (15 знаков) (5). Выполнение команды LEDB снова перезаписывает первоначальные данные. Первые 8 разрядов остаются свободными (6).

На светодиодном дисплее модулей центральных процессоров, имеющих соответствующее оснащение, можно отобразить цифры 0...9, все буквы между A и Z (только заглавные, без диакритических знаков), а также следующие символы: < > = x/' + -

### 7.9.1 PR

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

**Операнды MELSEC A**

	Операнды																	Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки		
	битовые							словные (16 бит)							Константы		Указатели				Уровень	M9012	M9010	
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K				H (16#)	P	I	N
s							●	●	●	●	●													
d	●																				● <sup>1</sup>			

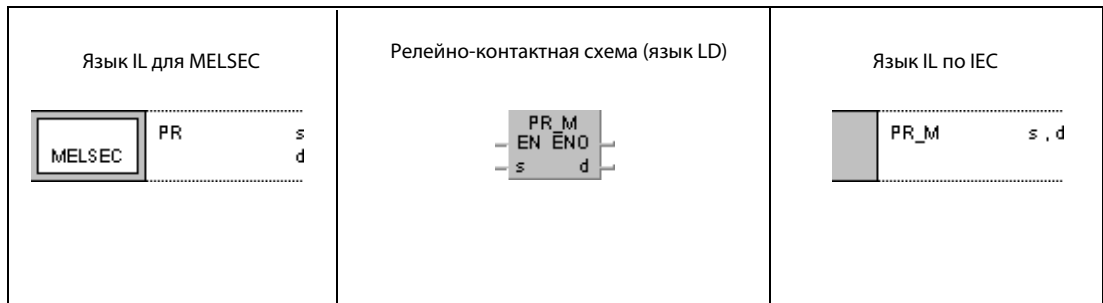
<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

**Операнды MELSEC Q**

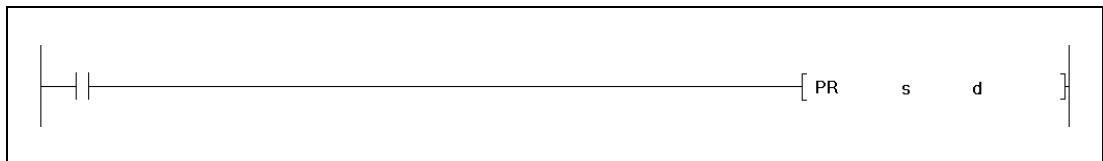
	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непос. адр. J□□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы §			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	●	●	—	—	
d	● <sup>1</sup>	—	—	—	—	—	●	—	—	3	

<sup>1</sup> только Y

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранен ASCII-код.	строковая величина
d	Начальный адрес модуля вывода, на который выводится ASCII-код.	бит

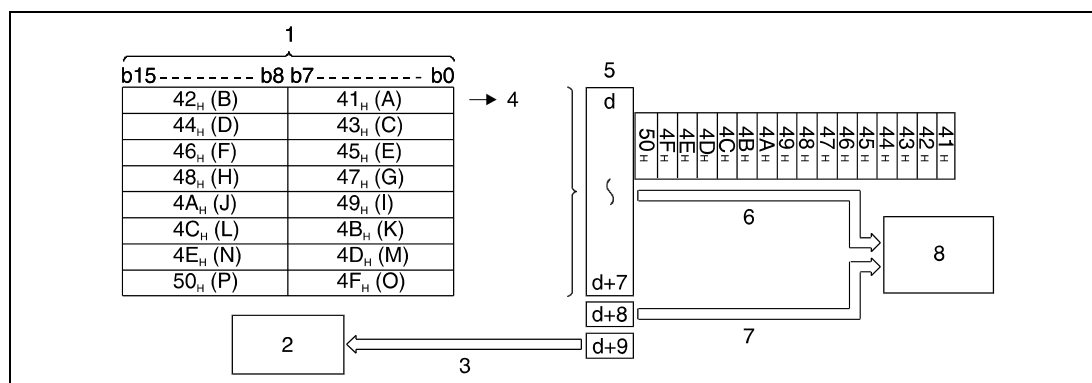


**Принцип действия****Вывод на периферийное устройство****PR Вывод строковой величины ASCII**

Команда PR имеет две возможные функции. Ее функция зависит от состояния специального маркера M9049 (серия "A") или маркера SM701 (серии "Q"/"System Q").

Маркер M9049/SM701 установлен (1) (функция 1):

Вывод 16-значной строковой величины в кодировке ASCII на выходной модуль. Строковая величина, распределенная на 2 группы по 8 знаков, считывается из области адресов s и выводится на выходы, указанные в d.



1 Операнд, в котором сохранен ASCII-код.

2 подпрограмма

3 флаг для индикации обработки команды PR (применяется в целях блокировки)

4 начало вывода

5 выходы Y

6 вывод ASCII-кода

7 вывод стробирующего сигнала

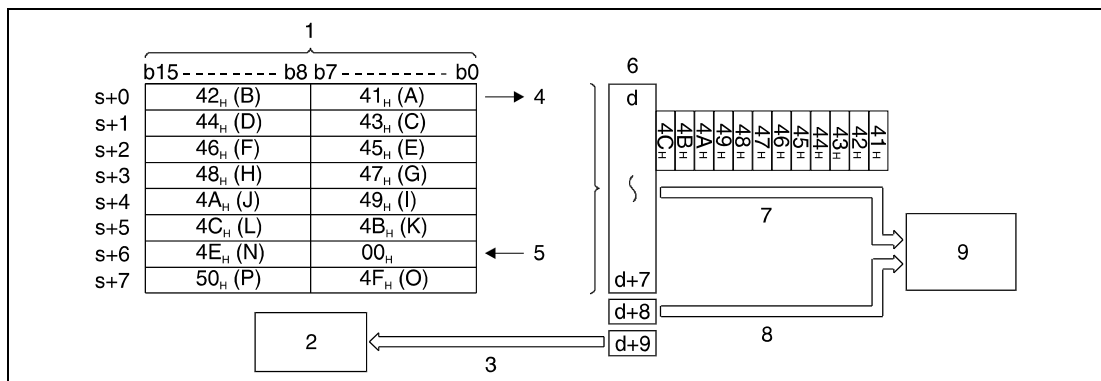
8 принтер или дисплей

Команда PR может использовать только данные, уже сохраненные в формате ASCII. Если сохраненные данные изменяются, выводятся текущие данные. Для конвертации буквенно-числовых данных в ASCII-код применяется команда ASC.

Во время передачи ASCII-кода, состоящего из 16 знаков, по адресу d + 9 устанавливается флаг выполнения команды PR. Это означает, что выход Y с адресом d + 9 остается включенным до тех пор, пока происходит выполнение команды PR.

M9049/SM701 не установлен (0) (функция 2):

Вывод строковой величины в виде шестнадцатеричных кодов ASCII до кода "00н" из области адресов  $s$  на выходы, указанные в  $d$ .



- 1 Операнд, в котором сохранен ASCII-код.
- 2 подпрограмма
- 3 флаг для индикации обработки команды PR (применяется в целях блокировки)
- 4 начало вывода
- 5 конец строковой величины (передача)
- 6 выходы  $Y$
- 7 вывод ASCII-кода
- 8 вывод стробирующего сигнала
- 9 принтер или дисплей

**ПРИМЕЧАНИЕ** В процессорах серии "А" возможна только 1-я функция.

Если во время вывода перезаписывается содержимое адресов операндов, в которых хранится ASCII-код, то выводятся новые данные.

Конец строковой величины ASCII обозначен знаком с кодом "00н".

Если в указанном операнде нет шестнадцатеричного кода "00н", обработка прерывается и устанавливается флаг ошибки.

Во время передачи ASCII-кода, по адресу  $Y = d + 9$  устанавливается флаг выполнения команды PR.

Для выполнения команды PRC нужен выходной модуль с 10 следующими друг за другом бинарными выходами. Диапазон адресов начинается с заданного в  $d$  адреса выхода.

Скорость обработки выходного сигнала в выходном модуле составляет 30 мс/знак. Таким образом, для обработки  $n$  знаков необходимо время  $n \times 30$  мс. Так как управление передачей при выводе происходит на основе прерываний с интервалами по 10 мс, программа работает непрерывно.

Обработка 10 выходных адресов в выходном модуле происходит независимо от обновления состояния входов и выходов после команды END в программе.

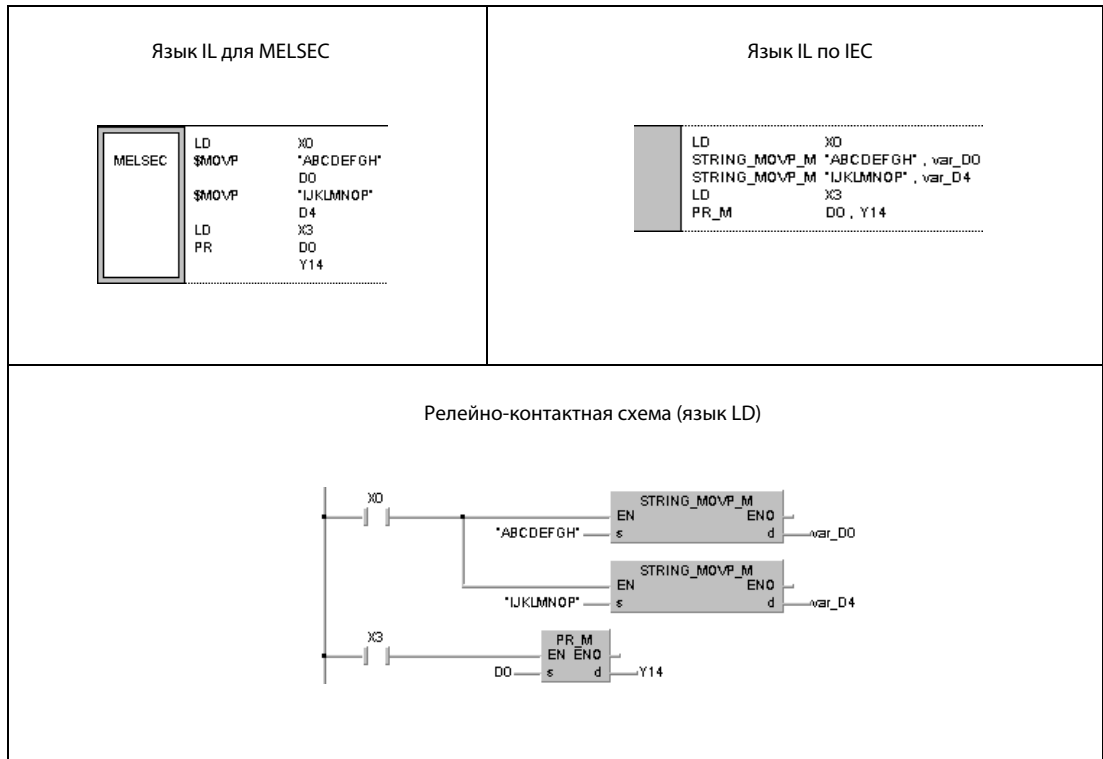
В дополнение к ASCII-коду через адрес  $Y = d + 8$  выводится стробирующий сигнал (включен 10 мс, выключен 20 мс).

Команды PR и PRC можно выполнять многократно, одну за другой. Однако программировать следует так, чтобы было исключено одновременное выполнение двух или более команд PR/PRC. Поэтому с целью блокировки команды следует использовать флаг выполнения (выходной операнд в  $Y = d + 9$ ).

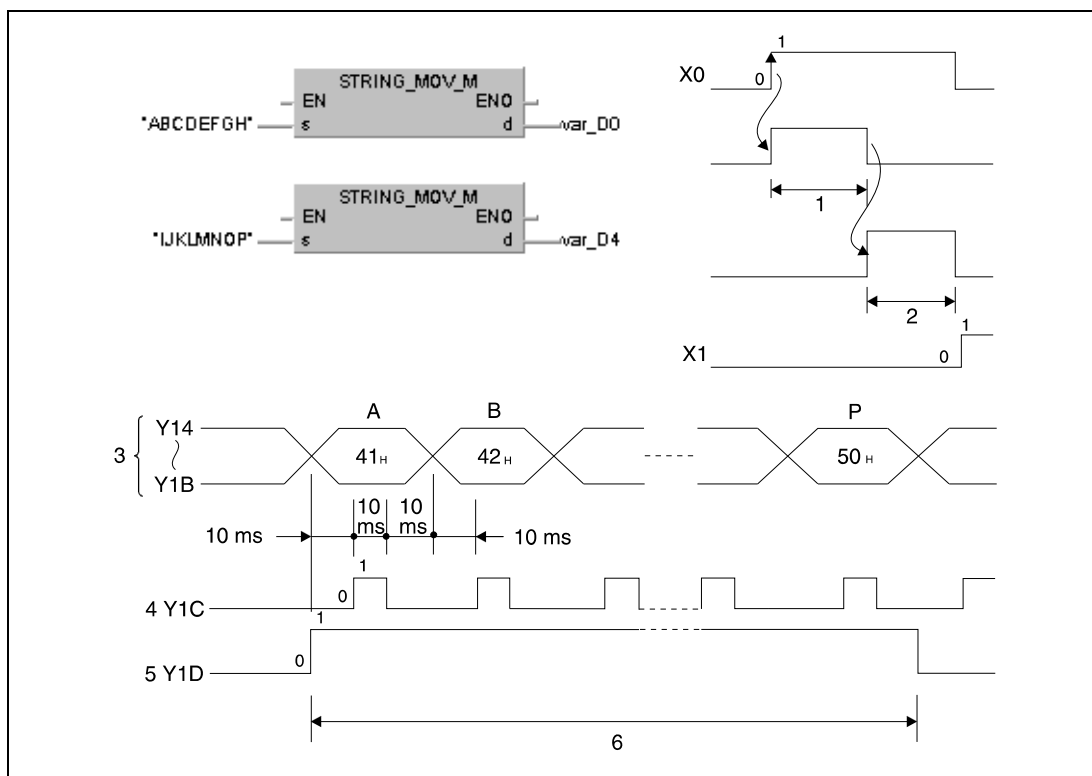
**Пример**

PR

Следующая программа при положительном фронте сигнала X0 преобразует строковую величину "ABCDEFGH" в ASCII-код и сохраняет его в регистрах данных с D0 по D7. После включения X3 ASCII-код выводится с адресов D0...D7 на выходы Y14...Y1D.



Принцип обработки программы наглядно пояснен на следующих диаграммах изменения сигналов.



- 1 сохранение строковой величины "ABCDEFGH" в D0...D3
- 2 сохранение строковой величины "IJKLMNOP" в D4... D7
- 3 ASCII-код
- 4 стробирующий сигнал
- 5 флаг обработки команды PR
- 6 обработка команды PR (длительность = 480 мс)

**ПРИМЕЧАНИЯ** Если используется иной кроме процессор серии "A", и маркер SM701 не установлен, в регистр D8 необходимо записать значение "00h". Отсутствие этого кода в вышеприведенном примере привело бы к возникновению ошибки.

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства. Компилятор или проверяющая программа могут выдать сообщения об ошибках.

### 7.9.2 PRC

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

**Операнды MELSEC A**

	Операнды																	Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9011				
	битовые							словные (16 бит)							Константы		Указатели						Уровень			
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K							H (16#)	P	I
s	●	●	●	●	●	●	●	●	●	●	●									●	●					
d	●																							7 <sup>1</sup>	●	

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

**Операнды MELSEC Q**

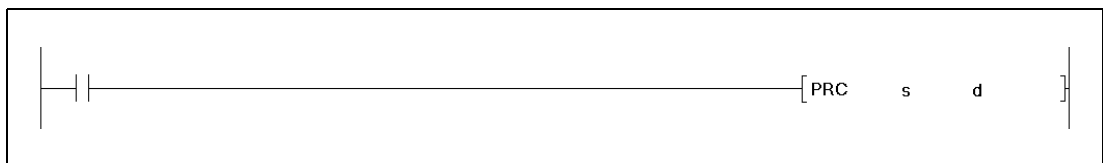
	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы \$	Иные P, I, J, U		
	битовые	словные		битовые	словные						
s	●	●	●	●	●	—	—	●	—	—	3
d	● <sup>1</sup>	—	—	—	—	—	—	—	—	—	

<sup>1</sup> только Y

**GX IEC Developer**



**GX Developer**



**Переменные**

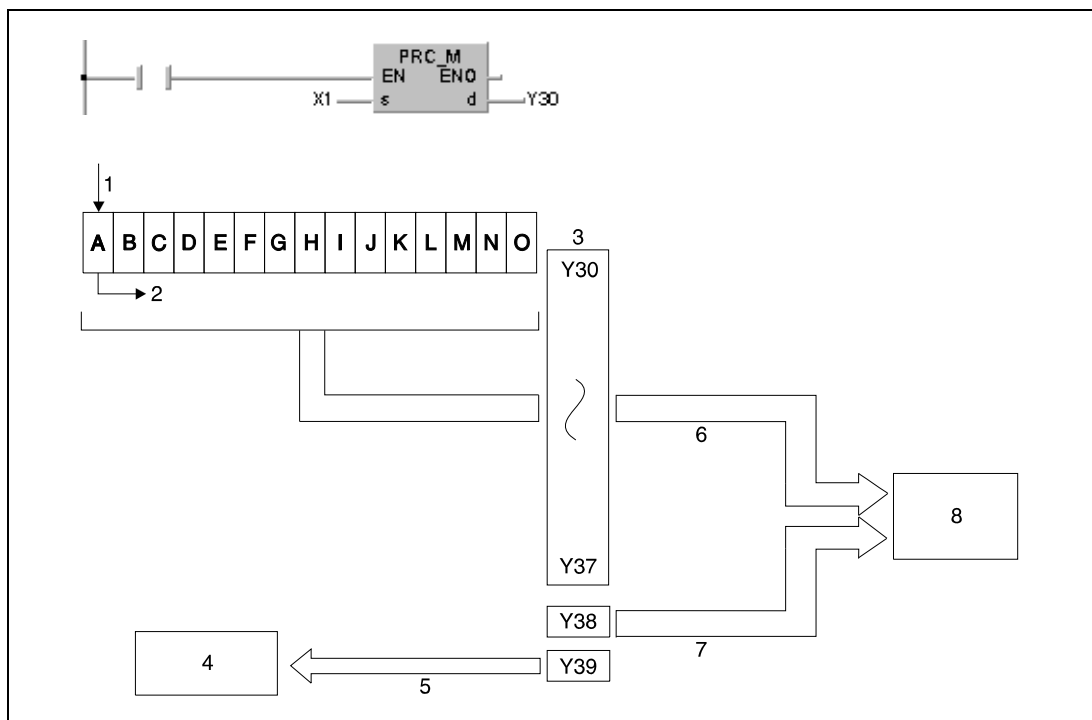
Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранен выводимый комментарий.	BIN, 16 бит
d	Адрес модуля вывода, по которому выводится комментарий.	бит

**Принцип действия**      **Вывод на периферийное устройство**  
**PRC**      **Вывод комментария**

Команда PRC выводит комментарий операнда (в ASCII-кодировке) на выходной модуль.

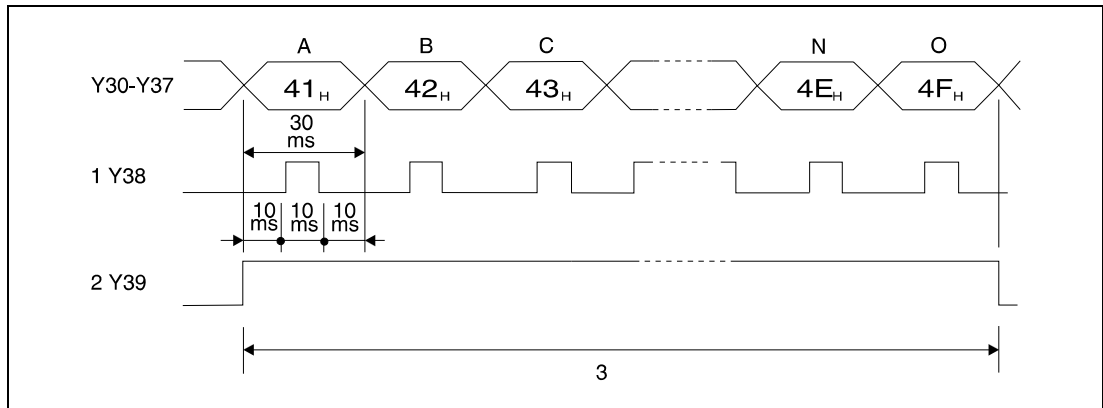
В контроллерах MELSEC серии "A" из области адресов *s* считывается строковая величина, распределенная на 2 группы по 8 знаков, и выводится на выходы, указанные в *d*.

В процессорах серий "Q" и "System Q" имеется возможность выбора между выводом 16 или 32 знаков. Выбор осуществляется с помощью специального маркера SM701. Если маркер SM701 установлен (1), выводятся 16 знаков. Если SM701 не установлен (0), выводятся 32 знака.



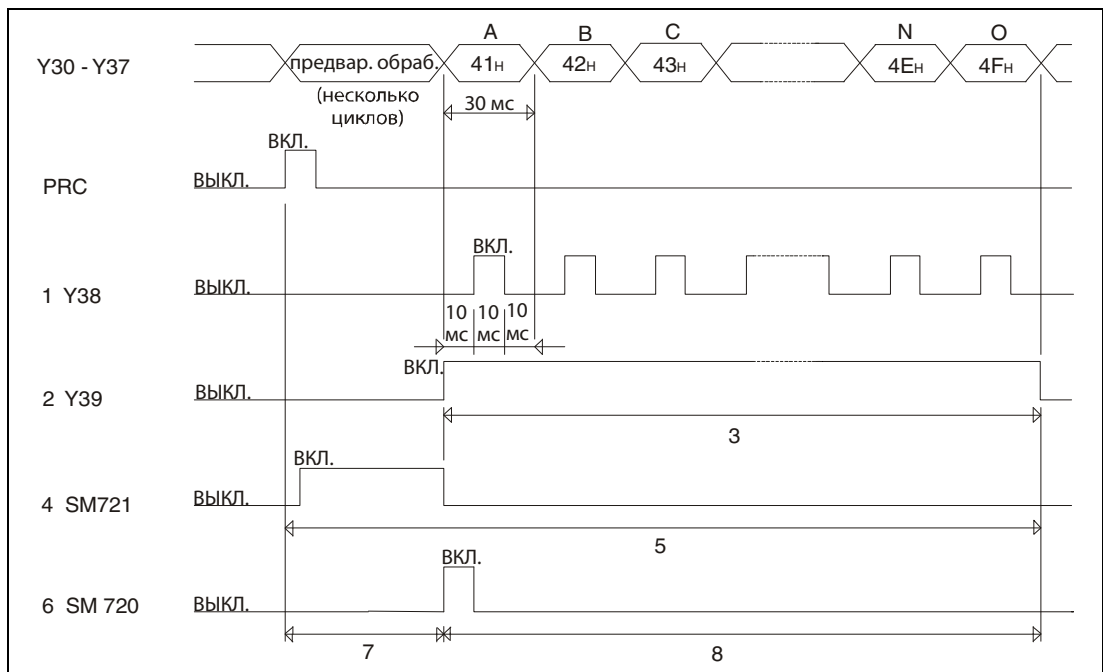
- 1 комментарий (ASCII-код), начиная с X1
- 2 начало вывода
- 3 выходы Y
- 4 подпрограмма
- 5 флаг для индикации обработки команды PR (применяется в целях блокировки)
- 6 вывод ASCII-кода
- 7 вывод стробирующего сигнала
- 8 принтер или дисплей

Принцип обработки команды PRC наглядно пояснен на следующих диаграммах изменения сигналов.  
 Диаграмма изменения сигналов в случае процессоров QnA:



- 1 стробирующий сигнал
- 2 флаг обработки команды PRC
- 3 обработка команды PRC (длительность = 16 x 30 мс = 480 мс)

Диаграмма изменения сигналов в случае многопроцессорного ЦПУ серии "Q":



- 1 стробирующий сигнал
- 2 флаг вывода сигнала
- 3 обработка команды PRC (длительность = 16 x 30 мс = 480 мс)
- 4 флаг обращения к файлу
- 5 В это время другая команда PRC не может выполняться.
- 6 флаг завершения обращения к файлу
- 7 В это время никакие другие команды не могут выполняться.
- 8 Все команды кроме PRC, S.FREAD, S.FWRITE, PLOAD, PUNLOAD и PSWAPP могут выполняться.



Используются 10 бинарных выходов цифрового выходного модуля. Диапазон адресов начинается с адреса выхода Y, заданного в d.

Скорость обработки выходного сигнала в выходном модуле составляет 30 мс/знак. Таким образом, для обработки n знаков необходимо время  $n \times 30$  мс. Так как управление передачей при выводе происходит на основе прерываний с интервалами по 10 мс, программа работает непрерывно.

В дополнение к ASCII-коду, через адрес  $Y = d + 8$  выводится стробирующий сигнал (включен 10 мс, выключен 20 мс).

Во время передачи ASCII-кодов, состоящих из 16 знаков, по адресу  $d + 9$  устанавливается флаг выполнения команды PRC, т. е. выход Y с адресом  $d + 9$  остается включенным до тех пор, пока происходит выполнение команды PRC. Команды PR и PRC можно выполнять многократно, одну за другой. Однако программировать следует так, чтобы было исключено одновременное выполнение двух или более команд PR/PRC. Поэтому с целью блокировки команды следует использовать флаг выполнения (выходной операнд в  $d + 9$ ).

Если в области адресов s нет данных, команда не выполняется.

Команда PRC может использовать только комментарии, уже сохраненные в контроллере. Для конвертации буквенно-числовых данных в ASCII-код применяется команда ASC.

После выполнения команды в процессорах серий "Q"/"System Q" на один цикл устанавливается специальный маркер SM720. Во время выполнения команды PRC устанавливается маркер SM721. Если SM721 уже установлен, команда PRC не может выполняться.

**ПРИМЕЧАНИЯ** *Комментарии, к которым обращается команда PRC, должны храниться на карте памяти. Использование комментариев во внутренней памяти не возможно.*

*Файл комментариев, к которому обращается команда PRC, устанавливается в режиме параметрирования в параметре "PC File Setting". Если файл не установлен, команда PRC не может выводить комментарии.*

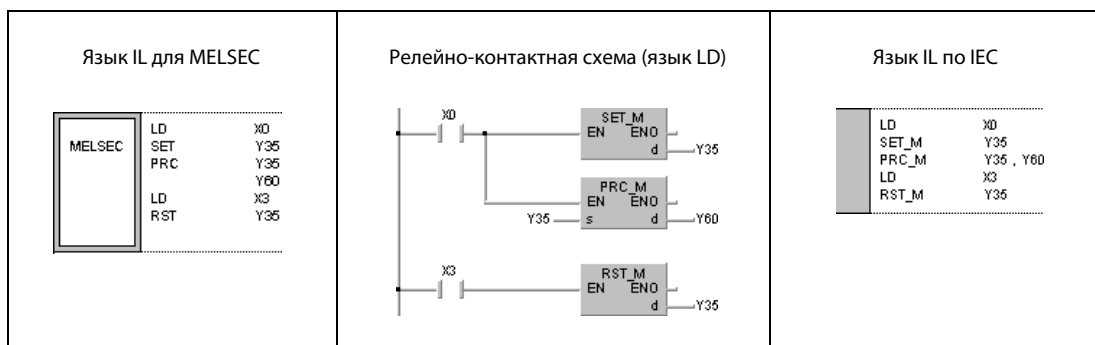
*Команду PRC нельзя выполнять в программе прерываний. Это может стать причиной неправильного функционирования.*

*Операнды комментариев для команды PRC сохраняются на карте памяти IC. Сохранять комментарии во внутренней памяти центрального процессора не возможно (только серия "Q").*

**Пример**

PRC

В следующей программе после включения входа X0 устанавливается выход Y35 и одновременно комментарий к этому выходу выводится в виде ASCII-кода на выходы Y60...Y69. После включения входа X3 выход Y35 сбрасывается.



### 7.9.3 LED

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
	● <sup>1</sup>	● <sup>2</sup>	●	● <sup>3</sup>	

<sup>1</sup> только для процессора A3N

<sup>2</sup> только для процессора A3A

<sup>3</sup> кроме процессора Q2A(S1)

**Операнды MELSEC A**

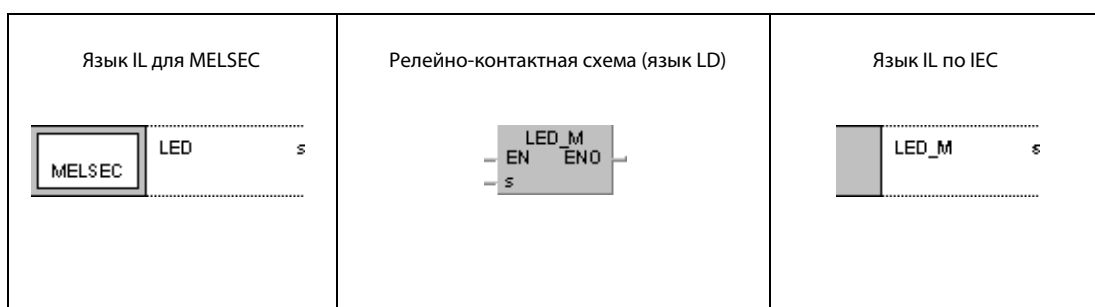
s	Операнды														Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011						
	битовые							словные (16 бит)												Константы		Указатели		Уровень	
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1						Z	V	K	H (16#)	P	I
							●	●	●	●	●											3 <sup>1</sup>	●		●

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

**Операнды MELSEC Q**

s	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы \$			Иные
	битовые	словные		битовые	словные						
	—	●	●	—	—	—	—	●	—	—	2

**GX IEC Developer**



**Переменные**

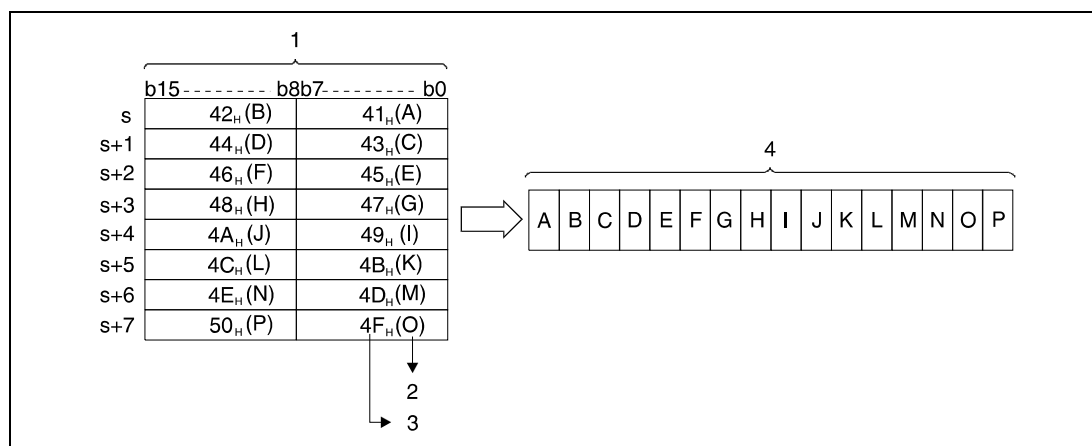
Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранены отображаемые данные в формате ASCII.	строковая величина

**Принцип действия**

**Вывод на светодиодный дисплей**

**LED Индикация данных, сохраненных в формате ASCII, на светодиодном дисплее центрального процессора**

Команда LED считывает данные в формате ASCII (16 знаков) из заданной области адресов и отображает их на светодиодном дисплее процессорного модуля, имеющего соответствующее оснащение. Начальный адрес ASCII-кодов, сохраненных в 8 адресах, задается в *s* (см. следующую схему).



<sup>1</sup> отображаемые данные

<sup>2</sup> знак ASCII

<sup>3</sup> код ASCII (шестнадцатеричный)

<sup>4</sup> светодиодный дисплей с передней стороны процессорного модуля

Если в указанной области адресов нет данных в формате ASCII, то индикация таймеров, счетчиков, регистров данных и регистров связи остается пустой. В случае регистров файлов R индикация имеет произвольное содержимое. Если соответствующие регистры файлов уже стерты, индикация остается пустой.

На светодиодном дисплее модулей центральных процессоров, имеющих соответствующее оснащение, можно отобразить цифры 0...9, все буквы между A и Z (только заглавные, без диакритических знаков), а также следующие символы: < > = x/ + -

Команда LED может использовать только данные, уже сохраненные в формате ASCII. Для конвертации буквенно-числовых данных в кодировку ASCII следует применять команду \$MOV или ASC.

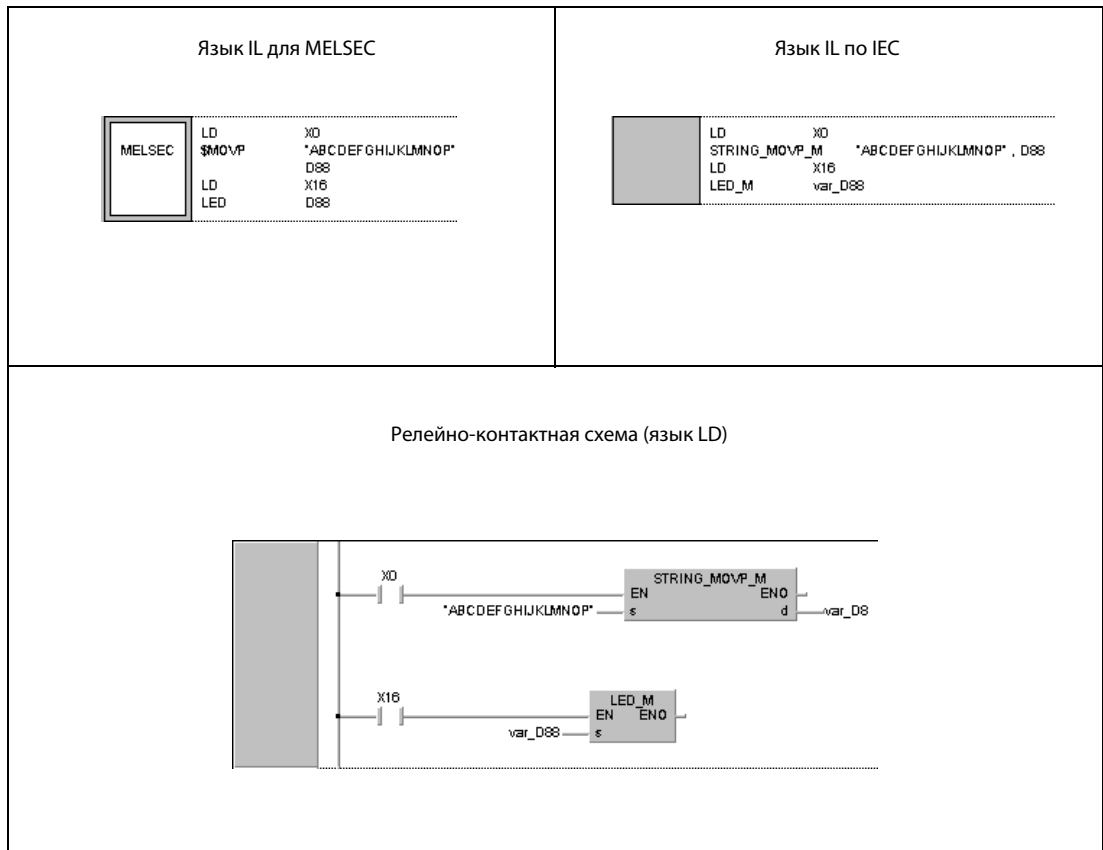
**ПРИМЕЧАНИЕ**

Команду LED могут выполнять только процессорах A3N, A3A, Q3A, Q4A и Q4AR. Если на модуле центрального процессора нет светодиодного дисплея, то выполнение этой команды никак не проявляется.

**Пример**

LED

Следующая программа преобразует строковую величину в ASCII-код, сохраняет этот код в заданных регистрах и выводит содержимое регистров на светодиодный дисплей центрального процессора. Для этого в первом шаге после включения X0 строковая величина "ABCDEFGHJKLMNOP" преобразуется в кодировку ASCII и сохраняется в регистрах данных с D88 по D95. После включения входа X16 данные в формате ASCII, сохраненные в регистрах D88...D95, отображаются на дисплее центрального процессора.



## 7.9.4 LEDC

## Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
	● <sup>1</sup>	● <sup>2</sup>	●	● <sup>3</sup>	

<sup>1</sup> только для процессора A3N

<sup>2</sup> только для процессора A3A

<sup>3</sup> кроме Q2A(S1)

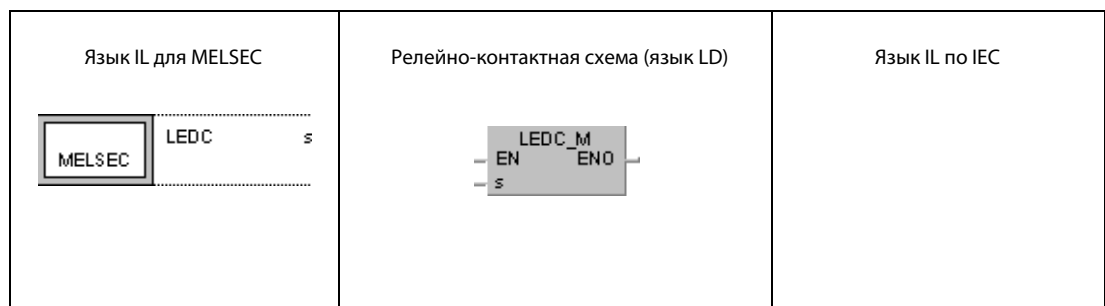
Операнды  
MELSEC A

Операнды																		Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9011					
битовые						словные (16 бит)						Константы		Указатели		Уровень											
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)						P	I	N		
●	●	●	●	●	●	●	●	●	●	●	●							●	●					3	● <sup>1</sup>		●

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

Операнды  
MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы \$	Иные			
битовые	словные		битовые	словные				BL, BLIS, BLTR			
s	●	●	●	●	●	—	—	●	—	—	2

GX IEC  
Developer

## Переменные

Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранен отображаемый комментарий.	адрес

**Принцип действия****Вывод на светодиодный дисплей****LEDC Индикация сохраненных комментариев на светодиодном дисплее центрального процессора**

Команда LEDC вызывает комментарии (16 знаков) из заданной области адресов и отображает их на светодиодном дисплее процессорного модуля, имеющего соответствующее оснащение. Если текст содержит более 16 знаков, отображаются только первые 16 знаков. Начальный адрес операндов, в которых хранятся комментарии, указывается в *s*.

Если указанный в *s* операнд не содержит комментария, дисплей с передней стороны процессорного модуля остается пустым. Если на нем уже была отображена какая-либо информация, она стирается. Если данные находятся вне области комментария, команда LEDC не обрабатывается и прежняя индикация сохраняется.

Если в комментарии имеются знаки, которые не могут быть отображены на дисплее центрального процессора, происходит ошибочная индикация. На светодиодном дисплее модулей центральных процессоров, имеющих соответствующее оснащение, можно отобразить цифры 0...9, все буквы между A и Z (только заглавные, без диакритических знаков), а также следующие символы: < > = x/ ' + -

Процессор Q2ACPU(S1) не способен выполнять команду LED. Ее обработка никак не проявляется.

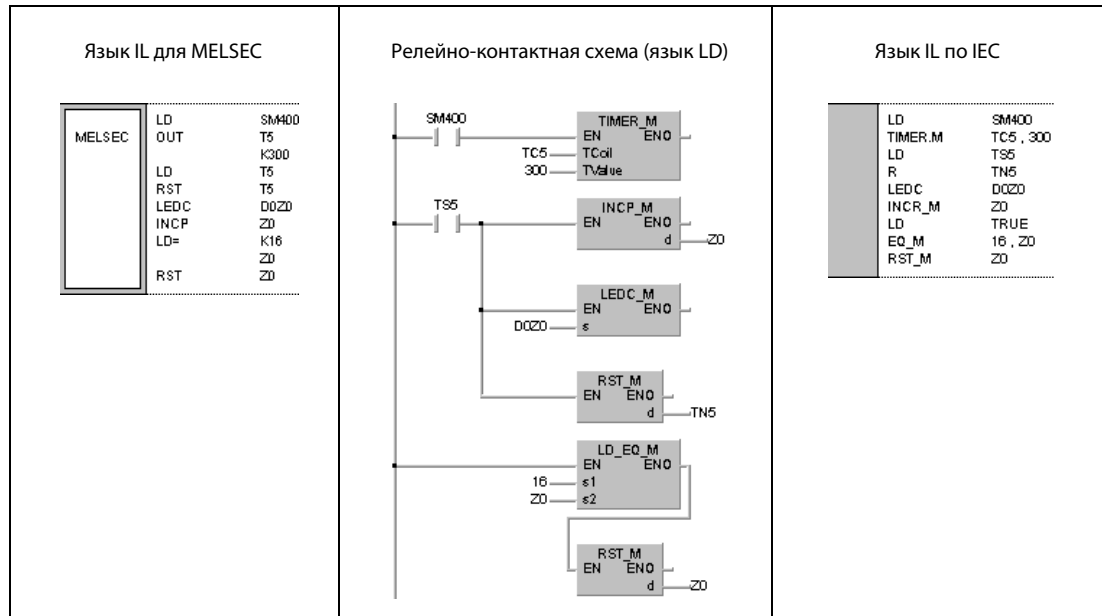
**ПРИМЕЧАНИЕ**

*В специализированных прикладных командах процессоров AnA команда LEDC служит для установки операндов. При программировании команды LEDC для процессора A3A необходимо соблюдать соответствующие указания, имеющиеся в отдельном руководстве по программированию процессоров AnA (раздел "Dedicated Instructions" или "Специализированные команды") (только серия "A").*

## Пример

## LEDC

В следующем примере программы на дисплей с 30-секундным интервалом выводятся комментарии из регистров D0...D15. Для этого таймер T5 с 30-секундным тактом включает входное условие для команды LEDC. Как только таймер включает контакт, отображается комментарий из регистра данных D (0 + Z). При каждом включении значение Z повышается на 1. Как только Z достигает 16, значение Z снова устанавливается на 0.





### 7.9.5 LEDA, LEDB

**Процессор**

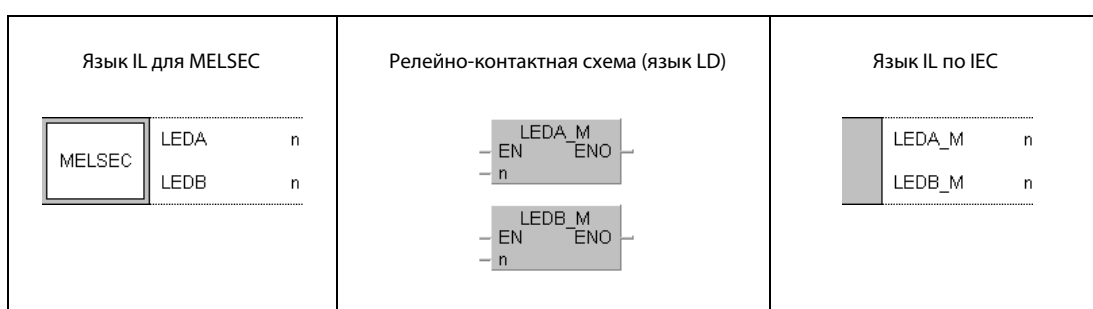
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
	● <sup>1</sup>				

<sup>1</sup> только для процессора A3N

**Операнды MELSEC A**

Операнды																Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки	
битовые						словные (16 бит)						Константы		Указатели							Уровень
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V				K	H (16#)	P
n																					

**GX IEC Developer**



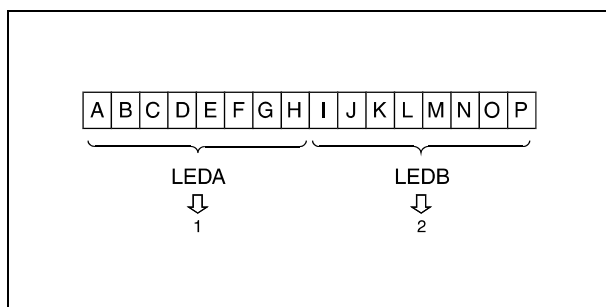
**Переменные**

Операнд	Значение	Тип данных
n	Данные в формате ASCII	строковая величина

**Принцип действия****Вывод на светодиодный дисплей****LEDA, LEDB Индикация строковой величины в кодировке ASCII на светодиодном дисплее центрального процессора**

Эти команды выводят строковую величину в кодировке ASCII на светодиодный дисплей модуля центрального процессора, имеющего соответствующее оснащение. Строковая величина состоит из 8 знаков на каждую команду и указывается в команде LEDA/LEDB.

В общей сложности обе команды могут выводить строковую величину длиной до 16 знаков. При этом команда LEDA выводит первые 8 знаков (левая половина светодиодного дисплея), а команда LEDB – последние 8 знаков (правая половина дисплея).



<sup>1</sup> первые восемь знаков

<sup>2</sup> следующие восемь знаков

На светодиодном дисплее модулей центральных процессоров, имеющих соответствующее оснащение, можно отобразить цифры 0...9, все буквы между A и Z (только заглавные, без диакритических знаков), а также следующие символы: < > = x/ + –

**ПРИМЕЧАНИЕ**

В процессорах AnA и AnU команда LEDA/LEDB служит для обозначения начала специализированных прикладных команд. При программировании команды LEDA/LEDB для процессора AnA или AnU необходимо соблюдать указания, содержащиеся в отдельном руководстве по программированию центральных процессоров AnA/AnU (раздел "Dedicated Instructions" или "Специализированные команды").

**Пример****LEDA, LEDB**

Следующая программа после включения входа XС отображает строковую величину "ABCDEFGH IJKLMNOP" на светодиодном дисплее центрального процессора.

Язык IL для MELSEC	Релейно-контактная схема (язык LD)	Язык IL по IEC
<pre> MELSEC  LD      XС          LEDA   "ABCDEFGH"          LEDB   "IJKLMNOP" </pre>		

**ПРИМЕЧАНИЕ**

Примечание к индикации на модуле центрального процессора

Если первые 8 знаков перезаписываются командой LEDA, то вторая половина строковой величины из 16 знаков, отображенной с помощью команды LED, гаснет. Аналогичным образом, если последние 8 знаков строковой величины, отображенной с помощью команды LED, перезаписываются командой LEDB, то первая половина индикации гаснет.

### 7.9.6 LEDR

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

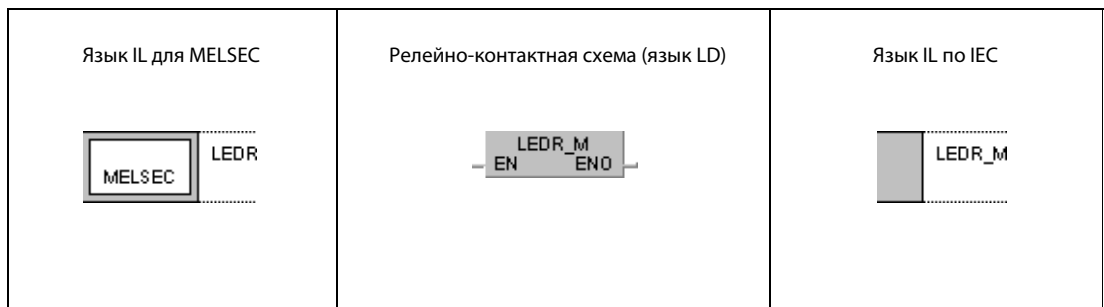
**Операнды MELSEC A**

Операнды															Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки					
битовые					словные (16 бит)					Константы		Указатели		Уровень										
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z				V	K	H (16#)	P	I	N	
																						1		

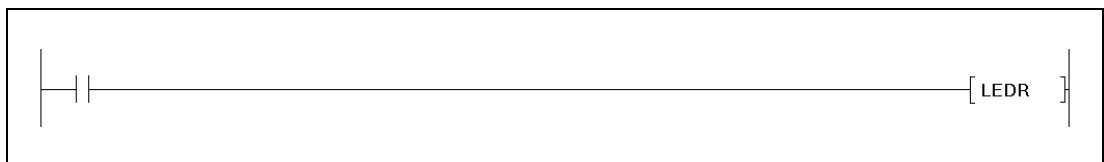
**Операнды MELSEC Q**

Операнды									Флаг ошибки	Шагов
Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы	Иные		
битовые	словные		битовые	словные						
—	—	—	—	—	—	—	—	—	—	1

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
—	—	—

**Принцип действия****Сброс маркеров ошибки и светодиодной индикации****LEDR Команда сброса**

Команда LEDR сбрасывает маркеры ошибки, автоматически устанавливаемые после возникновения ошибок. В модулях центральных процессоров со светодиодным дисплеем функция команды LEDR эквивалентна нажатию клавиши "INDICATOR RESET" (только серия "A").

**Действие команды LEDR в отношении маркеров ошибки, установленных при самодиагностике (только серия "Q"):**

Если в ходе самодиагностики выявлена ошибка, не влияющая на работу центрального процессора, то при выполнении команды LEDR гаснет светодиод "ERROR" и стирается индикация ошибок на светодиодном дисплее центрального процессора.

Дополнительно в прикладной программе необходимо сбрасывать маркеры SM1 и SD0, так как команда LEDR не сбрасывает их автоматически. Прочие шаги, связанные со сбросом маркеров ошибки, также не выполняются.

**Обработка команды LEDR при неисправности батареи (серия "Q"):**

Если после замены батареи выполняется команда LEDR, то светодиод "BAT. ARM" с передней стороны центрального процессора гаснет и индикация ошибок на светодиодном дисплее центрального процессора стирается. При этом автоматически сбрасывается SM51.

**Обработка команды LEDR при установленном маркере ошибки F в случае центрального процессора, не оснащенного светодиодным дисплеем:**

При выполнении команды LEDR происходят следующие процессы.

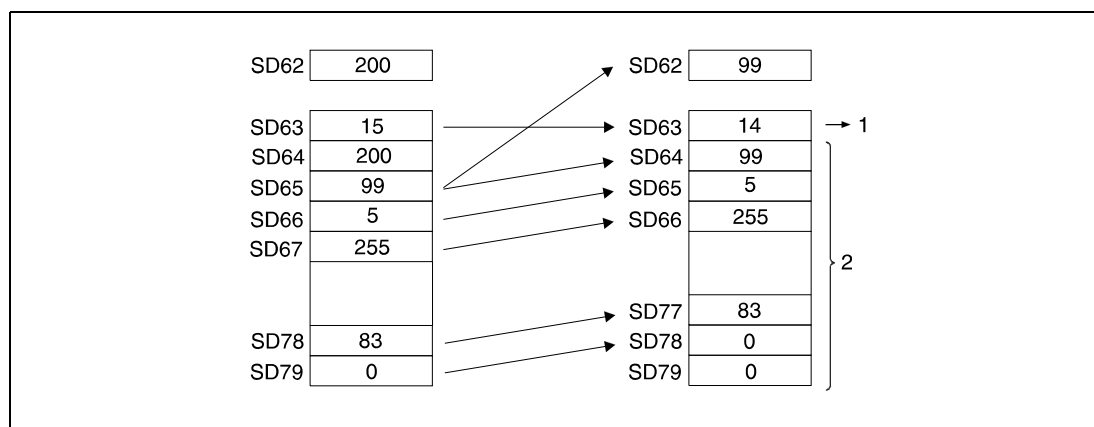
Светодиод "ERROR" на центральном процессоре мигает и через небольшое время гаснет.

Маркеры ошибок F, сохраненные в регистре D9009 (серия "A") или SD62 (серии "Q"/"System Q"), сбрасываются.

Регистры D9009 и D9125 (серия "A") или SD62 и SD64 (серии "Q"/"System Q") стираются и маркеры ошибок, сохраненные в регистрах D9126...D9131 (серия "A") или SD65...SD79 (серии "Q"/"System Q") сдвигаются вперед для дальнейшей обработки.

Адрес следующего маркера ошибки, сдвинутый вперед в D9125 (серия "A") или SD64 (серии "Q"/"System Q"), передается в D9009 (серия "A") или SD62 (серии "Q"/"System Q").

Сумматор маркеров ошибки в D9124 (серия "A") или SD63 (серии "Q"/"System Q") понижается на 1. Если регистр D9124 (серия "A") или SD63 (серии "Q"/"System Q") уже имеет значение 0, это значение сохраняется.



<sup>1</sup> количество сохраненных маркеров ошибок

<sup>2</sup> область памяти для хранения маркеров ошибок

### Обработка команды LEDR при установленном маркере ошибки F в случае центрального процессора со светодиодным дисплеем:

При выполнении команды LEDR происходят следующие процессы:

Маркер ошибки, отображенный на светодиодном дисплее центрального процессора, стирается.

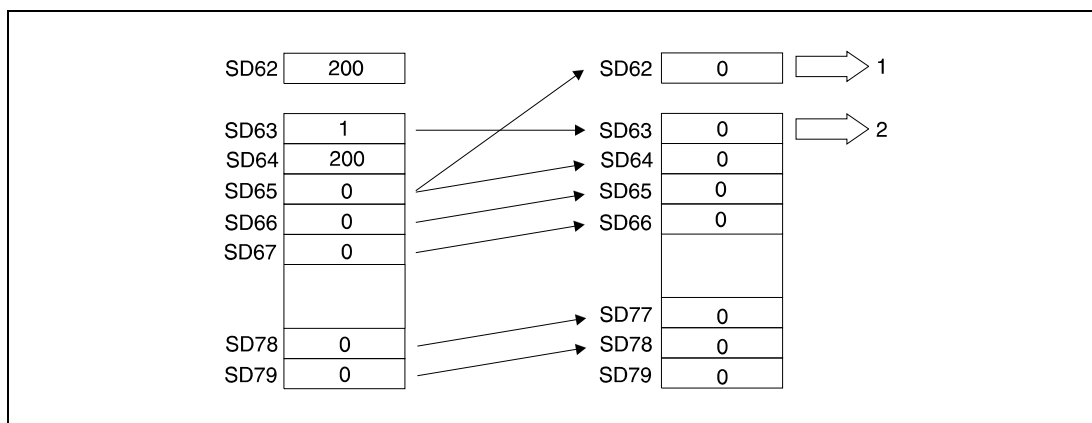
Маркер ошибки F, сохраненный в регистре D9009 (серия "A") или SD62 (серия "Q"), стирается.

Регистры данных D9009 и D9125 (серия "A") или SD62 и SD64 (серия "Q") сбрасываются, и маркеры ошибок, сохраненные в регистрах D9126...D9131 (серия "A") или SD65...SD79 (серия "Q"), сдвигаются вперед для дальнейшей обработки.

Адрес следующего маркера ошибки, сдвинутый вперед в D9125 (серия "A") или SD64 (серия "Q"), передается в регистр D9009 (серия "A") или SD62 (серия "Q").

Сумматор маркеров ошибки в D9124 (серия "A") или SD63 (серия "Q") понижается на 1. Если значение D9124 (серия "A") или SD63 (серия "Q") уже равно 0, это значение сохраняется.

На дисплее отображается маркер ошибки, находящийся теперь в D9009 (серия "A") или SD62 (серия "Q"). Если значение D9124 (серия "A") или SD63 (серия "Q") равно 0, индикация не происходит.



<sup>1</sup> Так как SD63 имеет значение 0, маркер ошибки не отображается на светодиодном дисплее.

<sup>2</sup> количество сохраненных маркеров ошибок

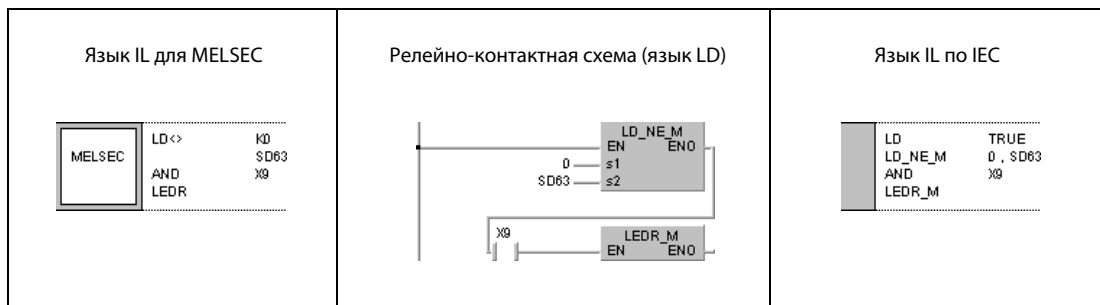
#### ПРИМЕЧАНИЕ

В процессорах AnA и AnU команда LEDR служит для завершения специализированных прикладных команд. При программировании команды LEDR для процессора A3A необходимо соблюдать соответствующие указания, имеющиеся в отдельном руководстве по программированию центральных процессоров AnA (раздел "Dedicated Instructions" или "Специализированные команды") (только серия "A").

**Пример**

## LEDR

В следующей программе выполняется команда LEDR, если значение в регистре SD63 не равно 0 и если установлен вход X9.

**ПРИМЕЧАНИЕ**

В следующей таблице перечислены предварительные настройки для номеров ошибок и приоритеты в регистрах диагностики с SD207 по SD209.

Приоритет	Номер ошибки (шестнадцатеричный)	Описание	Примечание
1	1	AC DOWN	Прервано электропитание
2	2	UNIT VERIFY ERR. FUSE BREAK OFF P. UNIT ERROR	Ошибка при проверке модулей ввода-вывода Сработал предохранитель Ошибка при проверке специального модуля
3	3	OPERATIN ERROR LINK PARA ERROR SFCP OPE. ERROR SFCP EXE. ERROR	Сбой при выполнении команды Ошибка в параметрах сети Ошибочная программа на языке SFC Сбой при выполнении программы на языке SFC
4	4	ICM.OPE ERROR FILE OPE ERROR EXTEND INST. ERROR	Сбой карты памяти Сбой при доступе к файлу Ошибка специализированной команды
5	5	PRG.TIME OVER	Превышено постоянное время цикла или сторожевой таймер
6	6	Команда CHK	Ошибка распознана с помощью команды CHK
7	7	Маркер ошибки	
8	8	Команда LED	
9	9	BATTERY ERROR	Неисправна батарея
10	A	Данные часов	

## 7.10 Выявление и устранение ошибок

Команды для выявления и устранения ошибок (отладки) служат для проверки на наличие ошибок, установки и сброса фиксации состояния, семплирования (выборочной трассировки) и контроля программы (трассировки). Следующая таблица содержит обзор команд.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту МЭК (IEC)
Контроль на ошибки	CHKST	CHKST_M
	CHK	CHK_M
	CHKCIR	CHKCIR_M
	CHKEND	CHKEND_MD
Установка и сброс фиксации состояния	SLT	SLT_M
	SLTR	SLTR_M
Установка и сброс семплирования (выборочной трассировки)	STRA	STRA_M
	STRAR	STRAR_M
Выполнение, установка и сброс контроля программы (трассировки)	PTRA	PTRA_M
	PtrAR	PtrAR_M
	PTRAEXE	PTRAEXE_M
	PTRAEXEP	PTRAEXEP_M

**ПРИМЕЧАНИЕ** *Пожалуйста, проверьте, поддерживает ли имеющаяся у вас версия среды GX IEC Developer эти команды.*

7.10.1 CHKST, CHK (серии "Q"/"System Q")

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

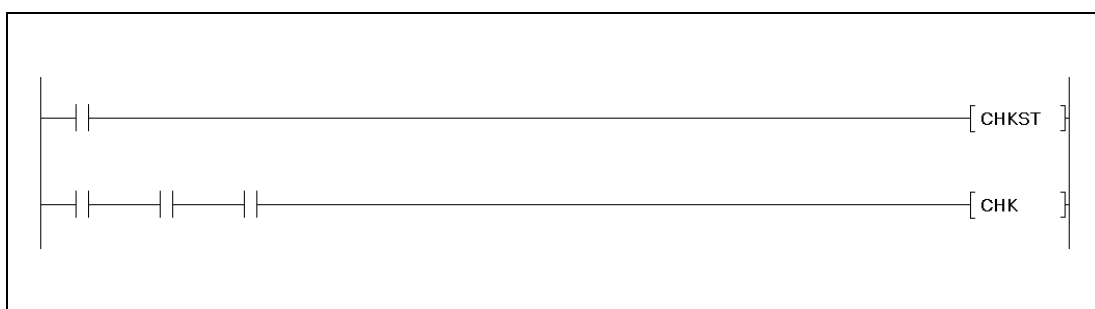
Операнды MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользov.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные				DY			
—	—	—	—	—	—	—	—	—	—	SM0	1

GX IEC Developer



GX Developer



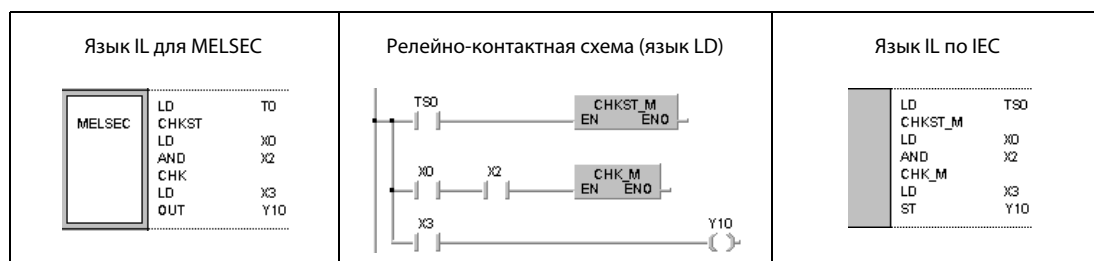
Переменные

Операнд	Значение	Тип данных
—	—	—



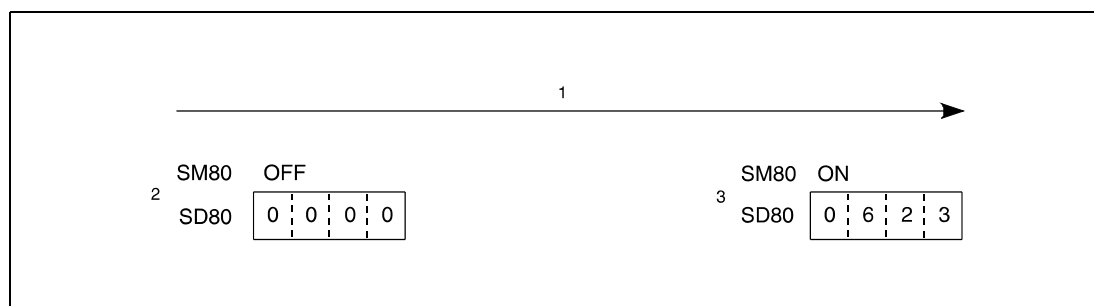
**Принцип действия****Проверка на ошибки при процессах переключения с изменением направления движения (серия "Q" и "System Q")****CHKST Начальная команда для команды CHK**

Команда CHKST запускает выполнение команды CHK. Если условие выполнения команды CHKST не установлено (0), выполняется очередной шаг программы, следующий за командой CHK. При установленном условии выполнения команды CHKST (1) выполняется команда CHK. Программирование этих команд показано на следующей иллюстрации.

**CHK Команда проверки на ошибки**

При использовании центральных процессоров некоторых типов (а также в зависимости от вида обработки) команда CHK позволяет проверять правильность расположения контактов конечных выключателей, контролирующих процессы движения со сменой направления. Если в расположении этих выключателей имеется ошибка, то устанавливается маркер SM80 и в регистр диагностики SD80 записывается соответствующий код ошибки.

При использовании процессора серии "Q" код ошибки сохраняется в виде 4-значного двоично-десятичного значения в регистре диагностики SD80. Три старших разряда отображают номер соответствующего контакта (в примере: контакт 62), а младший разряд – номер проверочного контура (состояние ошибки 1–6) (в примере: состояние ошибки 3).



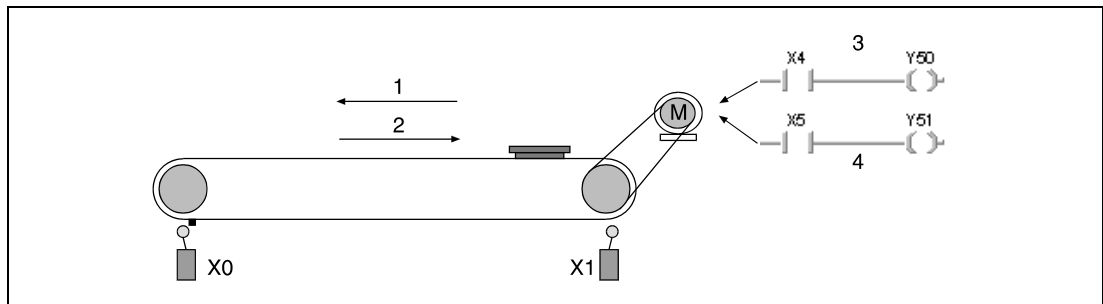
<sup>1</sup> контакт 62; катушка 3 (при распознавании ошибки)

<sup>2</sup> до распознавания ошибки

<sup>3</sup> после распознавания ошибки

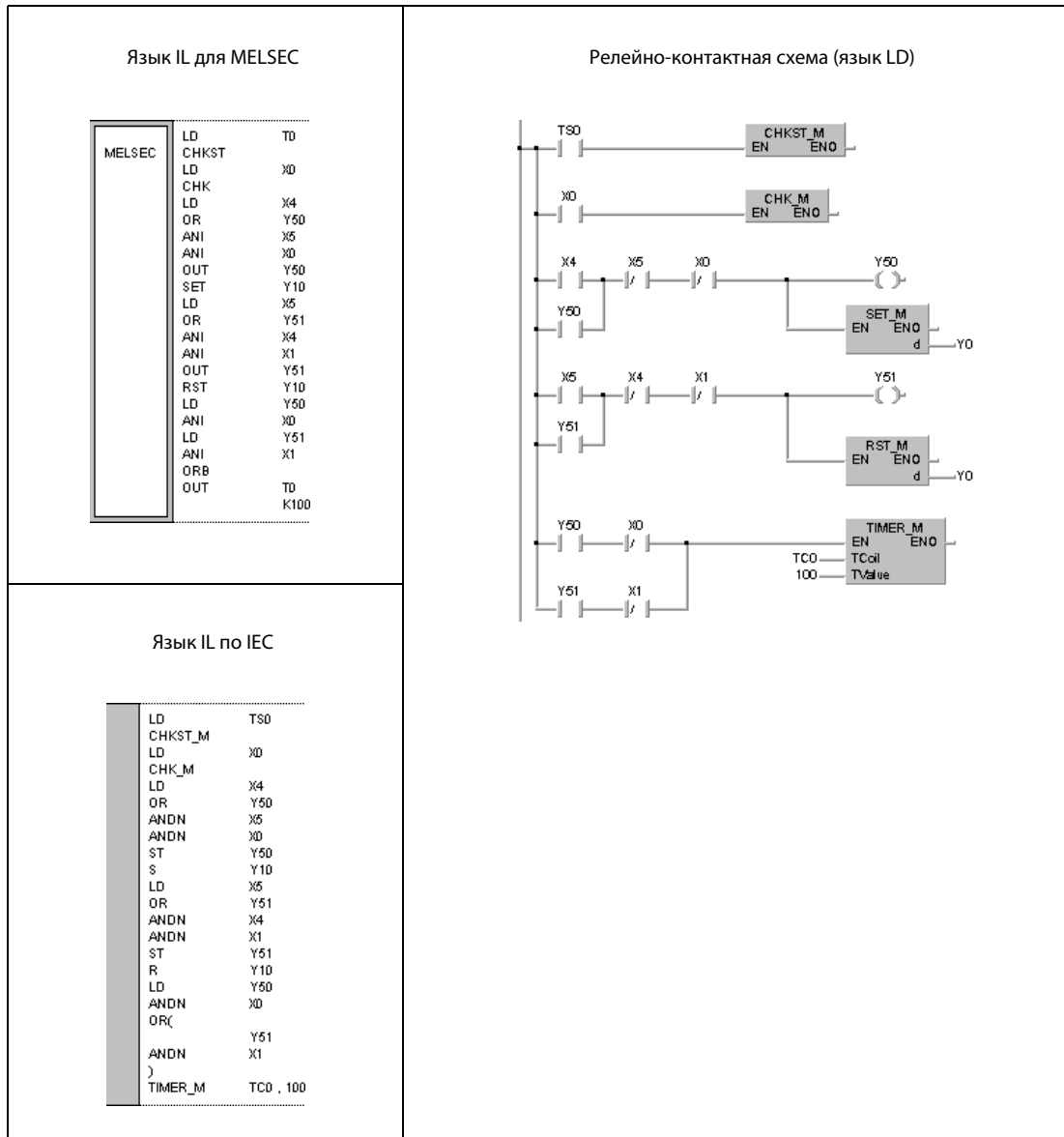
Входные контакты, программируемые перед командой CHK, не служат в качестве условия выполнения команды CHK. Они служат для задания условий контроля.

Чтобы наглядно разъяснить контроль с помощью команды CHK, ниже приведен пример конкретной задачи и соответствующей программы. На рисунке ниже изображен конвейер, который после достижения правого или левого предела начинает двигаться в противоположном направлении. Соответствующие конечные точки заданы концевыми выключателями (X0 и X1). В качестве пускового контакта для движения конвейера вперед используется X4, а для обратного движения – X5.



- 1 движение вперед
- 2 движение назад
- 3 запуск движения вперед
- 4 запуск движения назад

На рисунке ниже показана программа для управления вышеописанным конвейером и контроля на ошибки в случае применения процессора серии "Q". В случае безошибочной работы обработка переходит к шагу программы, следующему за командой CHK. При положительном фронте сигнала X4 конвейер начинает двигаться вперед и для контроля на ошибки устанавливается Y0. При положительном фронте сигнала X5 конвейер начинает двигаться назад и Y0 сбрасывается. Таймер T0 контролирует длительность рабочего цикла. При превышении времени устанавливается команда CHKST через контакт TS0. В следующем шаге программы выполняется команда CHK и в регистр диагностики SD80 записывается код ошибки.

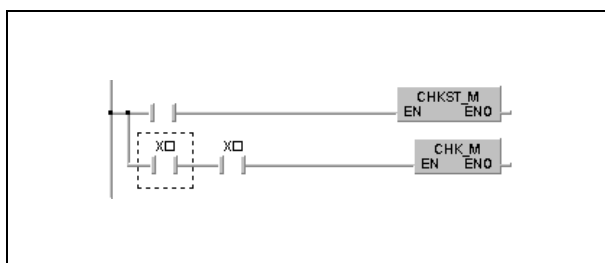


Принцип обработки команды CHK можно пояснить на основе следующих релейно-контактных схем, функция которых эквивалентна выполнению команды CHK. Адреса контактов концевого выключателя для движения вперед  $X□$  и концевого выключателя для движения назад  $X□ + 1$  должны быть присвоены подряд. Адрес концевого выключателя для движения вперед  $X□$  должен всегда быть ниже, чем адрес концевого выключателя для движения назад  $X□ + 1$ . Адрес контакта концевого выключателя для движения вперед  $X□$  присвоен выходу  $Y□$  с идентичным адресом. В соответствии с программой (см. пример), во время движения вперед этот выход установлен, а во время движения назад – сброшен.

Здесь, для улучшения наглядности предшествующего примера, контакты  $X0$  ( $X□$ ),  $X1$  ( $X□ + 1$ ) и  $Y0$  ( $Y□$ ) для задания контрольных условий вставлены непосредственно. В зависимости от программы, вместо них можно использовать любые другие адреса.

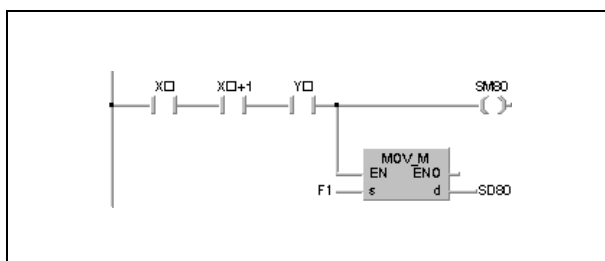
**ПРИМЕЧАНИЕ** *Выходы  $Y□$  обрабатываются в качестве внутренних маркеров и их невозможно опрашивать извне.*

Ниже показаны команды CHK. Контакт, обозначенный как  $X□$ , является переменной величиной, вмещающей максимум 150 контактов (150 конвейеров или подобных устройств).



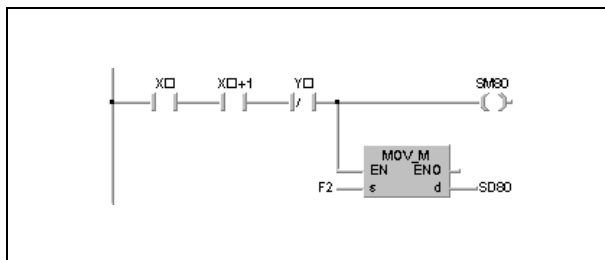
Проверочный контур 1 (состояние ошибки 1):

При движении конвейера вперед срабатывают оба концевых выключателя.



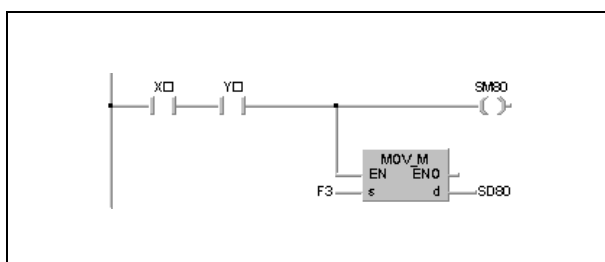
Проверочный контур 2 (состояние ошибки 2):

При движении конвейера назад срабатывают оба концевых выключателя.



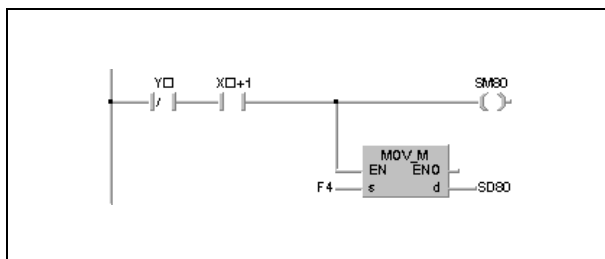
Проверочный контур 3 (состояние ошибки 3):

Вызов режима движения вперед при включенном концевом выключателе "Вперед".



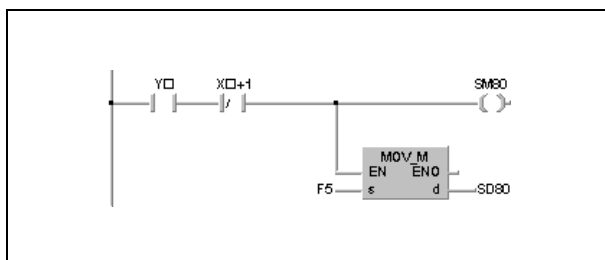
Проверочный контур 4 (состояние ошибки 4):

Вызов режима движения назад при включенном концевом выключателе "Назад".



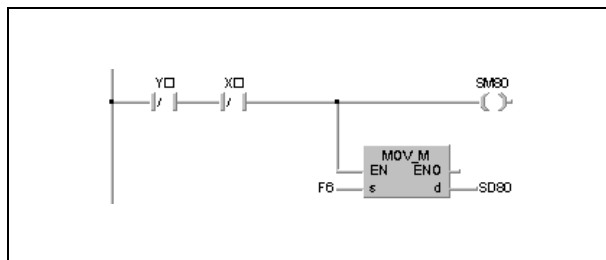
Проверочный контур 5 (состояние ошибки 5):

Вызов режима движения вперед при не включенном концевом выключателе "Назад".

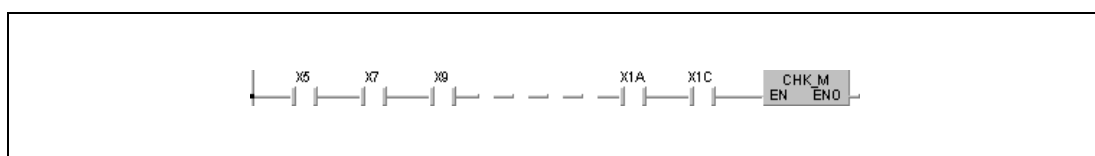


Проверочный контур б (состояние ошибки б):

Вызов режима движения назад при не включенном концевом выключателе "Вперед".



При использовании команды CHK можно указать максимум 150 адресов контактов для концевых выключателей движения вперед. При указании каждого очередного адреса пропускается один адрес для концевого выключателя движения назад.

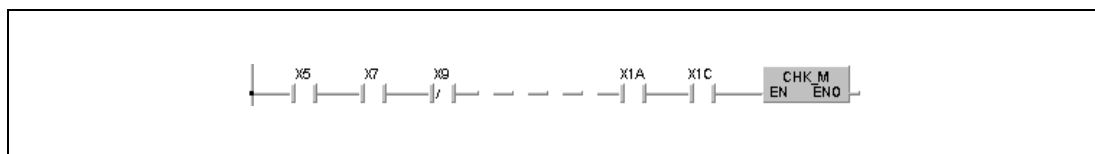


После выполнения команды CHK необходимо сбрасывать маркер SM80 и регистр диагностики SD80, так как после установки командой CHK они сохраняют свое состояние. Если при повторном выполнении команды CHK эти операнды не сброшены, выполнение команды CHK не возможно.

Перед командой CHK необходимо запрограммировать команду CHKST.

Команду CHK можно вписать в любой шаг основной программы. В одном программном компоненте (POU) команду CHK можно выполнить до двух раз.

Контрольные условия должны быть запрограммированы с помощью команды LD или AND перед командой CHK. Иные входные команды не возможны. Если в контрольных условиях программируется команда LDI или ANI, то контроль на ошибки с помощью команды CHK не возможен. Однако команды LDI и ANI могут обращаться к адресам контактов, участвующих в контроле на ошибки. На следующей иллюстрации выключатель с адресом X9 игнорируется, так как он представляет собой нормально-замкнутый контакт.



Алгоритм контроля зависит от состояния маркера SM710:

SM710 сброшен (0):

В этом случае контроль на ошибки происходит начиная с первого указанного адреса концевого выключателя с помощью первого проверочного контура. После того, как с помощью первого проверочного контура были проверены все адреса конечных выключателей для движения вперед, контроль на ошибки снова выполняется с первого адреса концевого выключателя, но уже с помощью второго проверочного контура. Контроль на ошибки завершается после того, как последний выключатель был проверен с помощью последнего (6-го) проверочного контура.

SM710 установлен (1):

Контроль на ошибки выполняется в последовательности входных контактов, запрограммированных в качестве контрольных переменных перед командой CHK. Это означает, что каждая схема, указанная адресом концевого выключателя для движения вперед, контролируется 6 проверочными контурами. Следующая схема контролируется лишь после завершения контроля всеми 6 проверочными контурами. Весь контроль на ошибки завершается после того, как последняя схема была проверена 6 проверочными контурами.

Если распознаны две ошибки или более, сохраняется только код ошибки с более высоким приоритетом. Так как ошибкой с самым высоким приоритетом является ошибка, найденная первой, сохраняется код самой первой ошибки.

#### **Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В контрольных условиях команды CHK два входных контакта включены параллельно (код ошибки 4235).
- Контрольные условия команды CHK содержат более 150 входных операндов (код ошибки 4235).
- За командой CHKST не запрограммирована команда CHK (код ошибки 4235).
- Команда CHK выполняется без предшествующей команды CHKST (код ошибки 4235).

### 7.10.2 CHK (серия "A")

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
● <sup>1</sup>	● <sup>1</sup>	●	●		

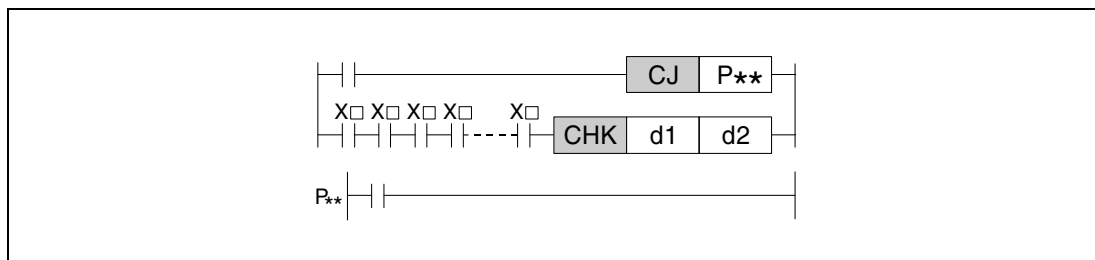
<sup>1</sup> только при непосредственной обработке

Операнды MELSEC A

	Операнды																	Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки	
	битовые							словные (16 бит)							Константы		Указатели						Уровень
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K				H (16#)	P	I
d1	●	●	●	●	●	●																	
d2	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●							5 <sup>1</sup>	

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

GX IEC Developer



Переменные

Операнд	Значение	Тип данных
d1	Операнд, устанавливаемый при выявлении ошибки.	бит
d2	Операнд, в котором сохраняется код ошибки.	BIN, 16 бит



**Принцип действия**      **Контроль на ошибки при процессах переключения с изменением направления движения (серия "A")**

**CHK      Команда контроля на ошибки**

Функция команды CHK зависит от выбранного вида обработки ввода-вывода. Если процессоры A1S и AnN используются в режиме внутреннего отображения состояний входов и выходов, команда CHK вырабатывает триггер.

При непосредственной обработке состояний входов и выходов (кроме процессоров AnA, AnAS, AnU и A2C) эта команда служит для контроля на ошибки при процессах переключения с изменением направления движения.

В связи с использованием указателя 254, программирование команды CHK возможно только на языке списка инструкций MELSEC MEDOC.

При использовании центральных процессоров некоторых типов (а также в зависимости от вида обработки) команда CHK позволяет проверять правильность расположения контактов конечных выключателей, контролирующих процессы движения со сменой направления. Если в схеме обнаружена ошибка, то устанавливается операнд в d1 и записывается соответствующий код ошибки в d2.

Входные контакты, программируемые перед командой CHK, служат не в качестве условия выполнения команды CHK, а для задания контрольных условий.

Задача команды CHK заключается в распознавании ошибок и выявлении их причины (например, превышение времени рабочего цикла). При безошибочном выполнении программы часть программы, содержащая команду CHK, должна пропускаться. Для пропуска этой части программы можно использовать команду CJ, SCJ или JMP.

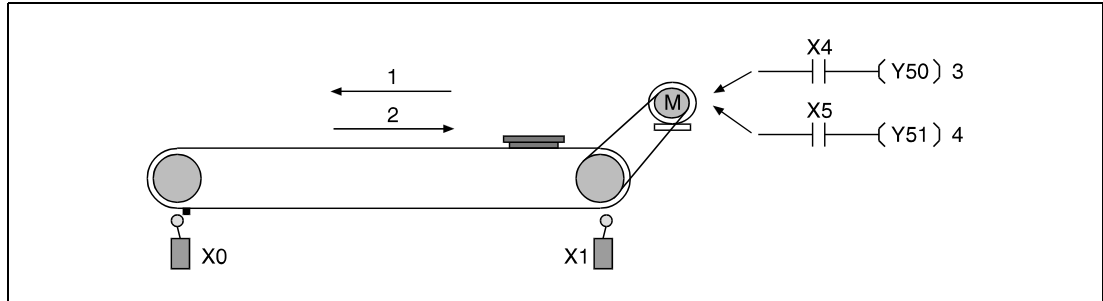
Команда CHK выполняется при каждом программном цикле и не зависит от состояния входных операндов, запрограммированных перед командой в качестве условий контроля.

Следующая программа при превышении времени рабочего цикла включает выход Y60 и выполняет команду CHK. После определения ошибки с помощью команды CHK устанавливается маркер M0 и происходит переход по адресу P31 (в примере этот адрес не показан). На метке перехода P31 может находиться, например, часть программы, запрограммированная для обработки ошибки. Если время не превышает, часть программы для контроля на ошибки пропускается, и выполняется шаг 18 по адресу перехода P30. В связи с использованием указателя 254, эта программа возможна только на языке списка инструкций MELSEC MEDOC.

100	LDI	Y060		
101	CJ	P30		
104	LD	M0		
105	CJ	P30		
108	P254			
109	LD	X010		
110	AND	X015		
111	AND	X008		
112	AND	X01A		
113	CHK	M0	D0	
118	P30			
118	LD	M10		
119	OUT	Y040		
121	END			

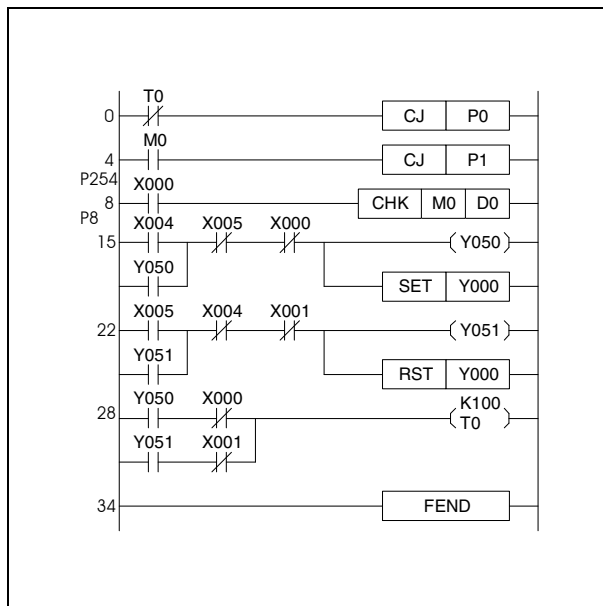


Чтобы наглядно разъяснить контроль с помощью команды СНК, ниже приведен пример конкретной задачи и соответствующей программы. На рисунке ниже изображен конвейер, который после достижения правого или левого предела начинает двигаться в противоположном направлении. Соответствующие конечные точки заданы концевыми выключателями (X0 и X1). В качестве пускового контакта для движения конвейера вперед используется X4, а для обратного движения – X5.

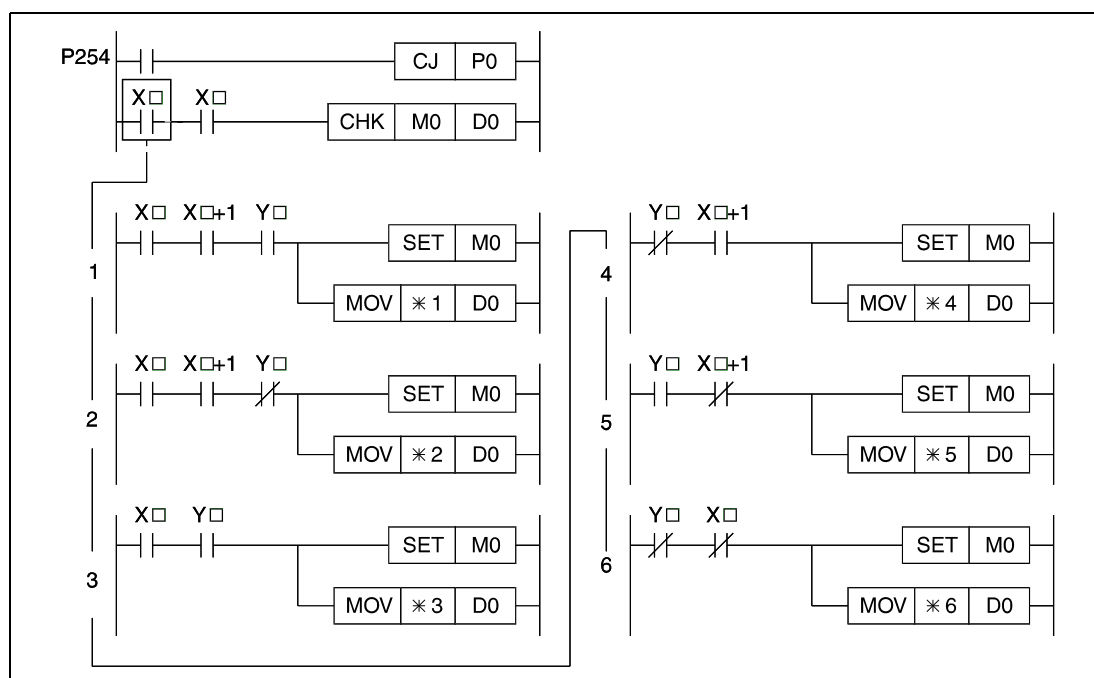


- 1 движение вперед
- 2 движение назад
- 3 запуск движения вперед
- 4 запуск движения назад

На рисунке ниже изображена программа для управления вышеприведенным конвейером и его контроля на ошибки. В связи с использованием указателя 254, эта программа возможна только на языке списка инструкций MELSEC MEDOC. При бесперебойной работе обработка переходит по адресу P0. После включения X4 конвейер начинает двигаться вперед и для контроля на ошибки устанавливается Y0. После включения X5 конвейер начинает двигаться назад и Y0 сбрасывается. Таймер T0 контролирует длительность рабочего цикла. При превышении времени команда СНК устанавливает маркер M0 и код ошибки сохраняется в D0. Программа продолжает выполняться для дальнейшей обработки ошибки по адресу перехода P1 (шаг 35).



Принцип обработки команды CHK можно пояснить с помощью следующей релейно-контактной схемы, функция которой эквивалентна выполнению команды CHK. Здесь, для улучшения наглядности предыдущего примера, контакты X0, X1 и Y0 для задания контрольных условий вставлены непосредственно. В зависимости от программы, вместо них можно использовать любые другие адреса.



Возможны следующие состояния ошибки:

Состояние 1: При движении конвейера вперед срабатывают оба концевых выключателя.

Состояние 2: При движении конвейера назад срабатывают оба концевых выключателя.

Состояние 3: Вызов режима движения вперед при включенном концевом выключателе "Вперед".

Состояние 4: Вызов режима движения назад при включенном концевом выключателе "Назад".

Состояние 5: Вызов режима движения вперед при не включенном концевом выключателе "Назад".

Состояние 6: Вызов режима движения назад при не включенном концевом выключателе "Вперед".

Номер кода ошибки, сохраненный в D0, соответствует номеру вышеуказанного состояния ошибки.

Контроль с помощью команды CHK происходит по показанному образцу. Этот образец релейно-контактной схемы не может быть изменен.

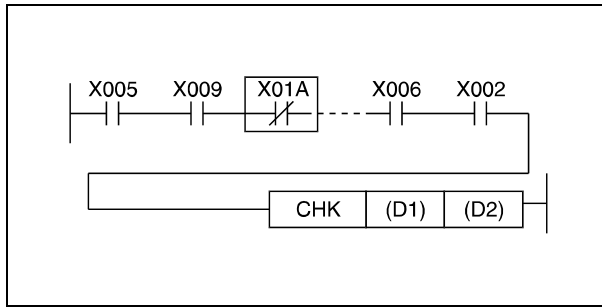
После выполнения команды CHK необходимо сбрасывать операнды в d1 и d2, так как после установки командой CHK они сохраняют свое состояние. Если при повторном выполнении команды CHK эти операнды не сброшены, выполнение команды CHK не возможно.

Перед строкой программы, содержащей команду CHK, должен всегда находиться указатель P254 в качестве адреса перехода. Этот указатель обозначает начало контроля на ошибки.

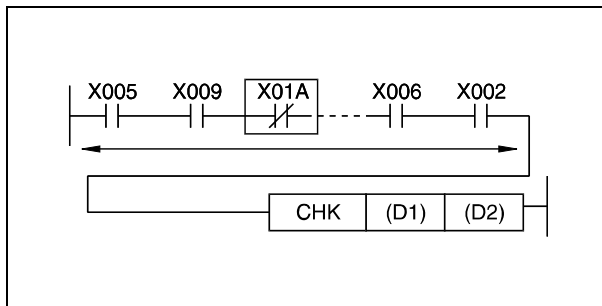
Команду CHK можно запрограммировать в любом шаге основной программы. Однако ее разрешается применять только один раз.

Команда CHK не может быть записана в режиме RUN центрального процессора.

Контрольные условия должны быть запрограммированы с помощью команды LD или AND перед командой CHK. Иные входные команды не возможны. Если в контрольных условиях запрограммирована команда ANI, контроль на ошибки с помощью команды CHK не возможен.



Контроль на ошибки выполняется в последовательности входных контактов, запрограммированных в качестве контрольных переменных перед командой CHK. Если распознаны две ошибки или более, сохраняется только код ошибки с более высоким приоритетом.



Коды ошибок, сохраненные в d2, зависят от различных факторов:

P254						
	1	2	3	4	5	6
Состояния ошибок	Входные контакты от 1 до 50	Входные контакты от 51 до 100	Входные контакты от 101 до 150			
Состояние ошибки 1 (код ошибки № 1)	$100 + (2 \times (\text{адр. конт.}) - 1)$	$400 + (2 \times (\text{адр. конт.}) - 1)$	$700 + (2 \times (\text{адр. конт.}) - 1)$			
Состояние ошибки 2 (код ошибки № 2)	$101 + (2 \times (\text{адр. конт.}) - 1)$	$401 + (2 \times (\text{адр. конт.}) - 1)$	$701 + (2 \times (\text{адр. конт.}) - 1)$			
Состояние ошибки 3 (код ошибки № 3)	$200 + (2 \times (\text{адр. конт.}) - 1)$	$500 + (2 \times (\text{адр. конт.}) - 1)$	$800 + (2 \times (\text{адр. конт.}) - 1)$			
Состояние ошибки 4 (код ошибки № 4)	$201 + (2 \times (\text{адр. конт.}) - 1)$	$501 + (2 \times (\text{адр. конт.}) - 1)$	$801 + (2 \times (\text{адр. конт.}) - 1)$			
Состояние ошибки 5 (код ошибки № 5)	$300 + (2 \times (\text{адр. конт.}) - 1)$	$600 + (2 \times (\text{адр. конт.}) - 1)$	$900 + (2 \times (\text{адр. конт.}) - 1)$			
Состояние ошибки 6 (код ошибки № 6)	$301 + (2 \times (\text{адр. конт.}) - 1)$	$601 + (2 \times (\text{адр. конт.}) - 1)$	$901 + (2 \times (\text{адр. конт.}) - 1)$			

<sup>1</sup> адрес контакта 1

<sup>2</sup> адрес контакта 50

<sup>3</sup> адрес контакта 51

<sup>4</sup> адрес контакта 100

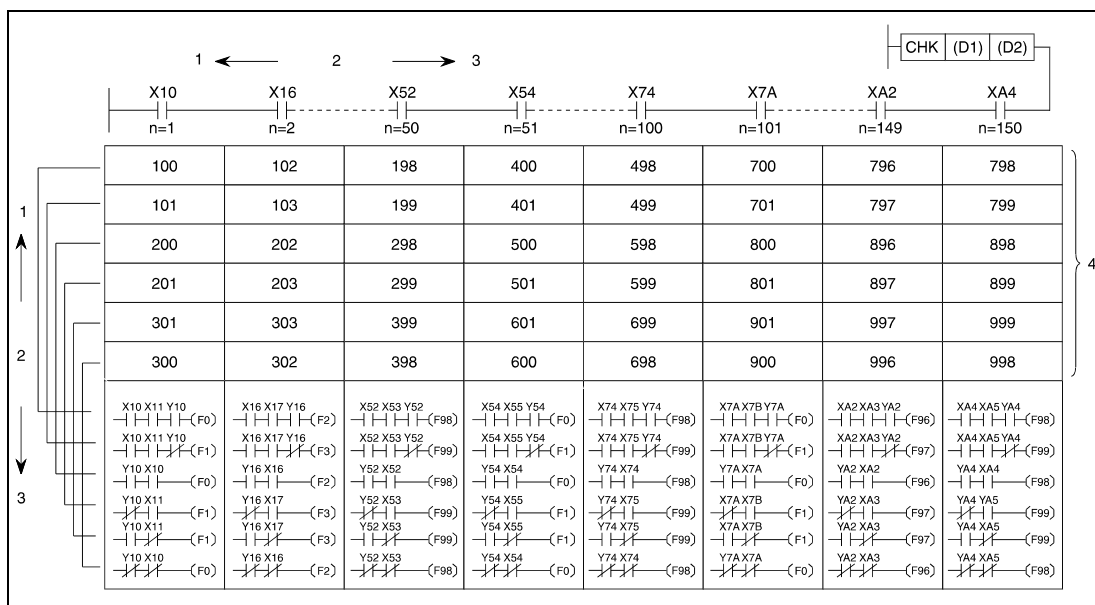
<sup>5</sup> адрес контакта 101

<sup>6</sup> адрес контакта 150

После выполнения команды СНК по выданным кодам ошибок можно судить о характере возникшей ошибки. Для быстрой диагностики неисправностей рекомендуется составить обзорную таблицу кодов ошибок.

Код ошибки	Причина	Устранение
301	Неисправность на конвейере 1: вызван режим обратного хода, хотя концевой выключатель для режима движения вперед не включен.	Проверить концевой выключатель X1, проверить конвейер
302	Неисправность на конвейере 1: ...	...
...	...	...

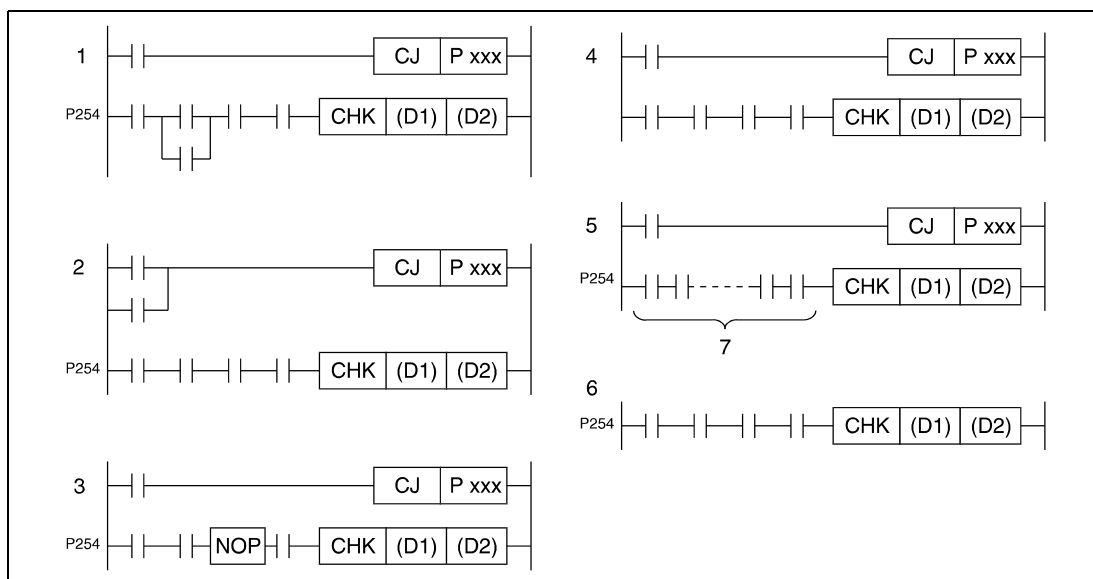
Обзор адресов кодов ошибок



- 1 высокий приоритет
- 2 приоритет
- 3 низкий приоритет
- 4 адрес кода ошибки

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки (данные в скобках относятся к иллюстрациям ниже):

- В контрольных условиях команды CHK (1) или в блоке команды CJ (2) два входных контакта включены параллельно.
- Среди контрольных условий команды CHK имеется команда NOP (3).
- В программе отсутствует адрес перехода P254 (4).
- Контрольные условия команды CHK содержат более 150 входных операндов (5).
- Перед блоком команд CHK отсутствует команда перехода (CJ) (6).



<sup>7</sup>более 150 контактов

### 7.10.3 CHKCIR, CHKEND

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

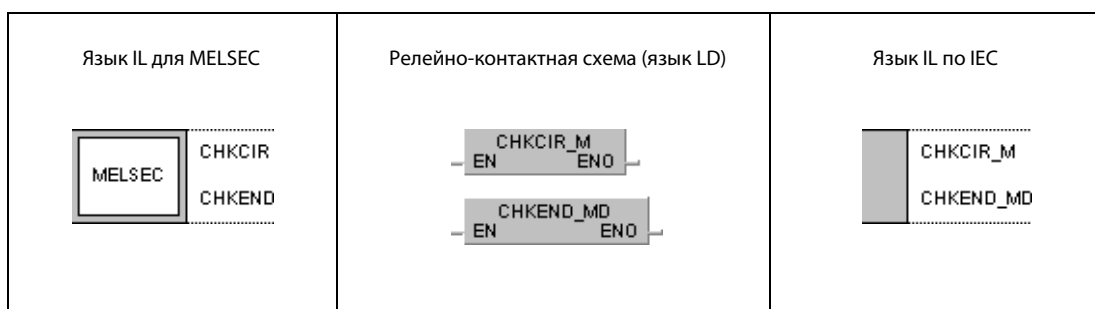
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R (только CHKEND).

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

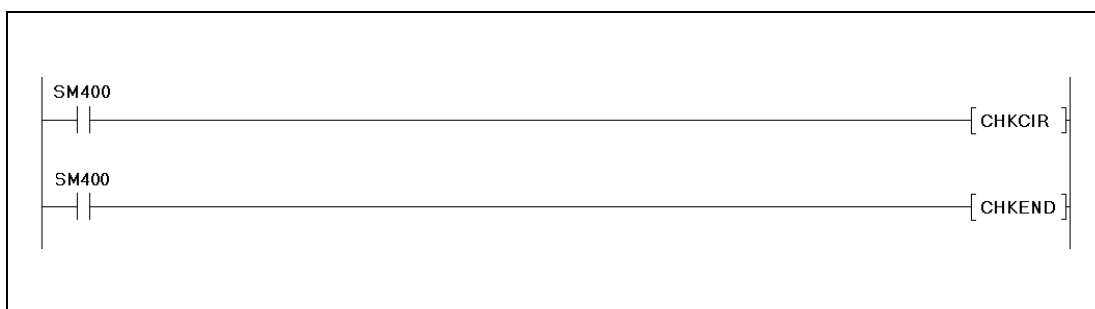
**Операнды MELSEC Q**

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы	Иные			
битовые	словные		битовые	словные							
—	—	—	—	—	—	—	—	—	—	S0	1

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
—	—	—



**Принцип действия**

**Создание проверочных контуров для команды CHK**

**CHKCIR, CHKEND**

**Команды начала и конца части программы с созданными проверочными контурами**

Команды CHKCIR и CHKEND служат для изменения проверочных контуров, используемых командой CHK. Можно создать проверочный контур любого требуемого формата. Сам контроль на ошибки выполняется с помощью команд CHKST и CHK.

Контроль на ошибки выполняется с помощью проверочных контуров, запрограммированных между командами CHKCIR и CHKEND.

**ПРИМЕЧАНИЯ**

*Если форматы проверочных контуров для команды CHK изменяются с помощью команд CHKCIR и CHKEND, необходимо запустить подключенные периферийные устройства в стандартном режиме, а затем выполнить процесс расширения программы.*

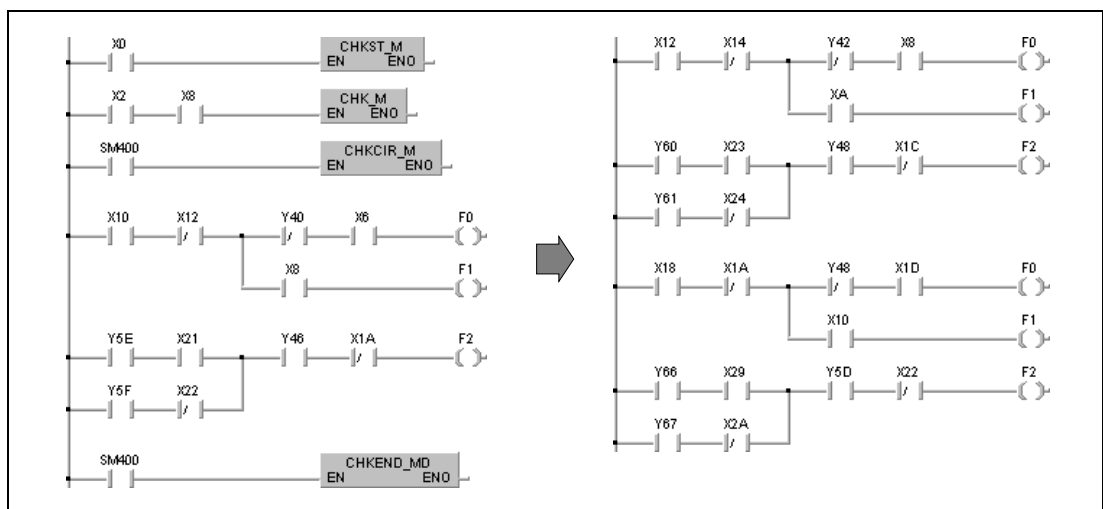
*Если периферийное устройство запускается со стороны процессора Q2A, Q2AS, Q3A или Q4A и предпринимается попытка создать измененные проверочные контуры для команды CHK с помощью команд CHKCIR и CHKEND, безупречная обработка более не обеспечивается.*

Новые проверочные контуры вырабатываются на основе имеющихся проверочных контуров, находящихся между командами CHKCIR и CHKEND, путем индексной адресации. Проверочным контурам, запрограммированным между этими командами, можно присвоить 9 маркеров ошибки (F1–F9). Индексная адресация выполняется путем сложения адресов контактов, указанных перед командой CHK, с адресами контактов проверочных контуров. Например, контакт X10 показанного ниже проверочного контура в индексированных проверочных контурах будет иметь обозначения X12 и X18, так как перед командой CHK запрограммированы контакты X2 и X8.

Алгоритм контроля на ошибки зависит от состояния маркера SM710 следующим образом:

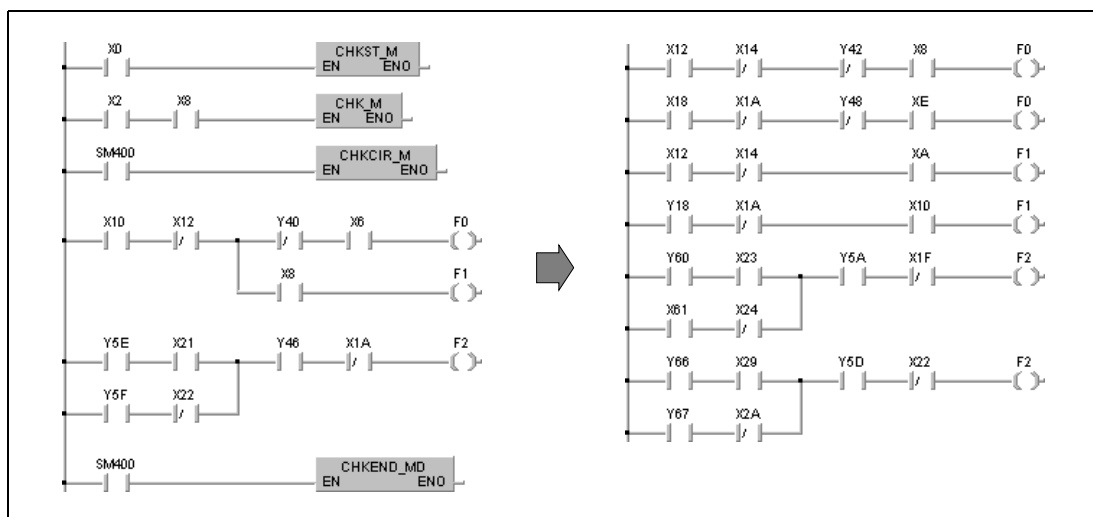
SM710 сброшен (0):

В этом случае каждый адрес контакта в проверочных контурах, запрограммированных между командами CHKCIR и CHKEND, сначала индексируется по первому адресу контакта, указанному перед командой CHK. Затем каждый запрограммированный проверочный контур индексируется по второму адресу контакта, указанному перед командой CHK. Этот процесс завершен, если для каждого запрограммированного проверочного контура с присвоенным маркером ошибки (F) существует количество новых проверочных контуров, эквивалентное входным контактам команды CHK.



SM710 установлен (1):

В этом случае сначала первый запрограммированный проверочный контур с присвоенным маркером ошибки индексируется по всем адресам контактов, запрограммированных перед командой CHK. Затем по всем адресам контактов, запрограммированным перед командой CHK, индексируется следующий проверочный контур. Этот процесс завершен, если для каждого запрограммированного проверочного контура с присвоенным маркером ошибки (F) существует количество новых проверочных контуров, эквивалентное входным контактам команды CHK.



Во время контроля на ошибки с помощью индексно адресованных проверочных контуров опрашивается состояние выходов (F), установленных с помощью команды OUT F. Если выход (F) установлен, устанавливается маркер SM80. В регистр диагностики SD80 записывается двоично-десятичный код ошибки, состоящий из номера контакта и проверочного контура (F1–F9).

При программировании проверочных контуров между командами CHKCIR и CHKEND можно использовать следующие команды.

Контакты:

LD, LDI, AND, ANI, OR, ORI, ANB, ORB, MPS, MPP, MRD и операторы сравнения.

Катушки:

OUT F

В качестве операндов для контактов следует программировать входы X и выходы Y.

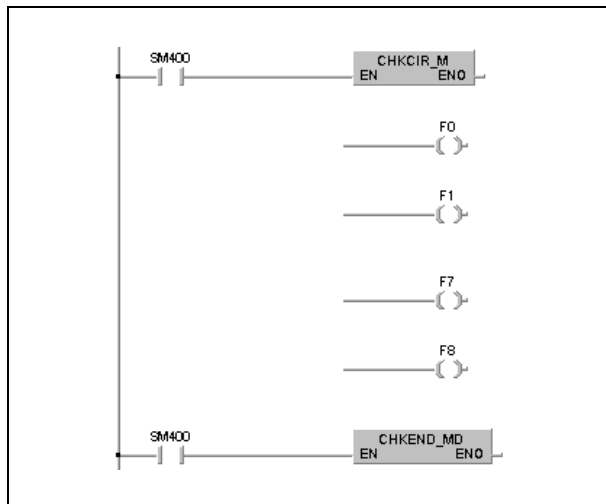
В качестве выходов проверочных контуров следует применять только маркеры ошибки (F). Проверочные контуры можно обозначать произвольно, начиная с маркера ошибки F0, так как эти выходы обрабатываются в качестве фиктивных контактов. Поэтому при обработке перекрывающихся маркеров ошибки (F) не возникают никакие проблемы.

Состояния маркеров ошибки (F) безупречно проверяются даже в том случае, если идентичный маркер ошибки (F) одновременно запрограммирован в иной взаимосвязи кроме команды CHK, так как оба этих маркера ошибки обрабатываются отдельно.

Так как состояние (0/1) маркеров ошибки (F), присвоенное командой CHK, не обновляется, маркеры ошибки (F) не устанавливаются даже в том случае, если они контролируются периферийным устройством.

Проверочные контуры, запрограммированные между командами CHKCIR и CHKEND, могут состоять из 256 шагов программы (контактных ветвей) и 9 выходов (маркеров ошибки F1–F9), используемых командами OUTF.

Проверочные контуры между командами CHKCIR и CHKEND обозначаются сверху вниз – от проверочного контура 1 (F0) до проверочного контура 9 (F8).



Команды CHKCIR и CHKEND можно запрограммировать на любом шаге основной программы. Однако в общей сложности их разрешается выполнять лишь два раза во всех программных файлах и не более одного раза в одном программном файле.

#### Источники ошибок

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Команды CHKCIR и CHKEND встречаются в проекте три раза или более (код ошибки 4235).
- После команды CHKCIR не выполняется команда CHKEND (код ошибки 4230).
- Выполняется команда CHKEND, хотя перед этим не была выполнена команда CHKCIR (код ошибки 4230).
- Программируются десять или более маркеров ошибки (F) (проверочных контуров) (код ошибки 4235).
- В проверочных контурах используются недопустимые операнды (код ошибки 4235).
- Один из операндов, запрограммированных в проверочных контурах, уже адресуется путем индексации (код ошибки 4235).

#### ПРИМЕЧАНИЯ

*Расширению программы могут воспрепятствовать следующие ошибки, возникшие в периферийном устройстве во время расширения программы:*

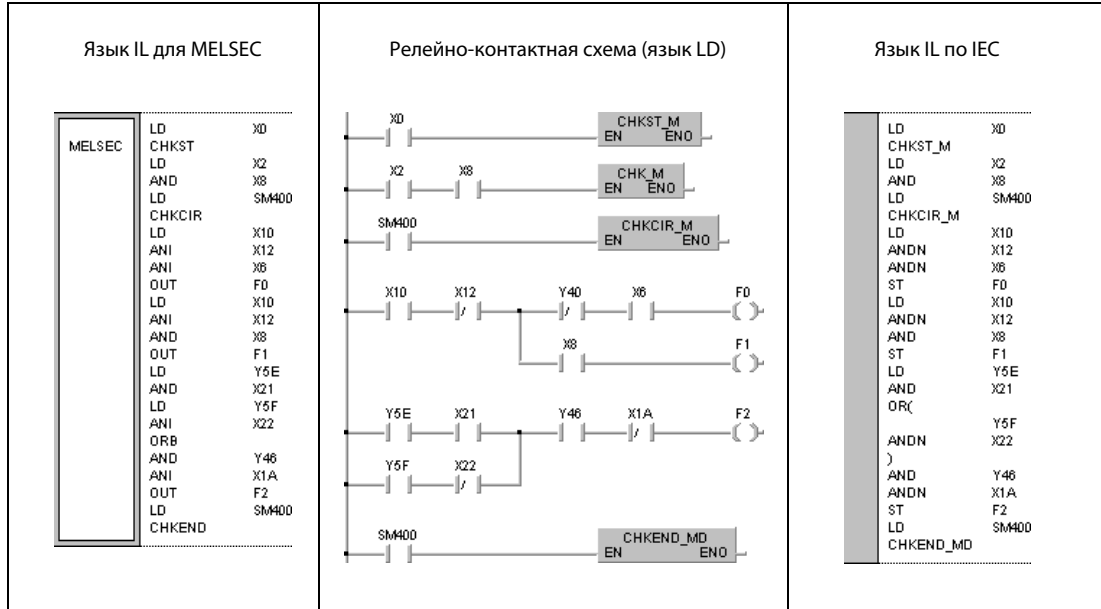
- *В проверочных контурах используются недопустимые операнды.*
- *Один из операндов, запрограммированных в проверочных контурах, уже адресуется путем индексации.*

*Если возникла одна из описанных здесь ошибок, проверочные контуры следует соответственно откорректировать.*

**Пример**

CHKCIR, CHKEND

Следующая программа выработывает индексно адресуемые проверочные контуры. Принцип действия этого примера программы разъяснен в пункте "Принцип действия". В дополнение к релейно-контактной схеме здесь приведены также программы на языке IL для MELSEC и стандартном языке IL по IEC.



### 7.10.4 SLT, SLTR

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	● <sup>1</sup>	●	●	●	

<sup>1</sup> кроме процессора A1N

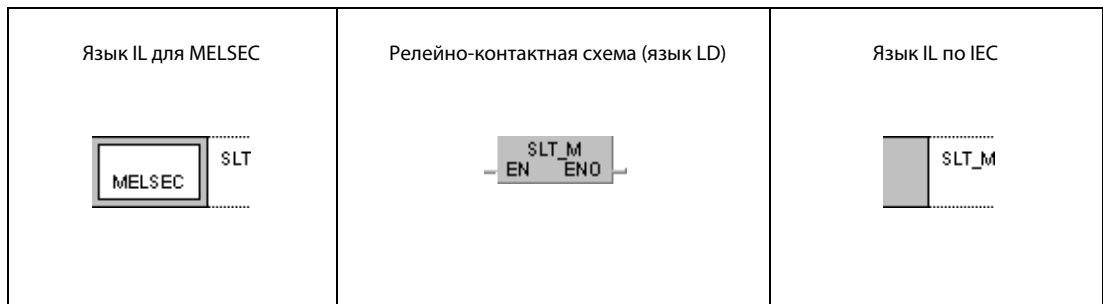
**Операнды MELSEC A**

Операнды															Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки						
битовые					словные (16 бит)					Константы		Указатели		Уровень											
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z				V	K	H (16#)	P	I	N		
																						1			

**Операнды MELSEC Q**

Операнды								Флаг ошибки	Шагов	
Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Z <sub>n</sub>	Константы			Иные
битовые	словные		битовые	словные						
—	—	—	—	—	—	—	—	—	—	1

**GX IEC Developer**



**Переменные**

Операнд	Значение	Тип данных
—	—	—

**Принцип действия****Фиксация и сброс фиксации состояния****SLT      Установка фиксации состояния**

Во время проверки программы с помощью среды GX IEC Developer не имеется возможности в любой момент передать и показать состояние любого операнда. Поэтому в памяти центрального процессора предусмотрена область для фиксации состояния (т. н. "память-защелка"). Эту область следует определить с помощью параметра. В ней сохраняются данные программного цикла (см. GX IEC Developer).

Команда SLT сохраняет в этой промежуточной памяти определенные данные операндов. Данные сохраняются в памяти фиксации состояния, после чего их можно проверять и отображать.

Команду SLT можно выполнять в программном цикле только один раз. Для очередного выполнения команду SLT необходимо сбрасывать с помощью команды SLTR.

**SLTR      Сброс фиксации состояния**

Команда SLTR стирает данные, сохраненные в промежуточной памяти фиксации, и сбрасывает команду SLT.

Повторное выполнение команды SLT возможно лишь после выполнения команды SLTR.

**ПРИМЕЧАНИЯ*****Пожалуйста, проверьте, поддерживает ли ваша версия среды GX IEC Developer функции фиксации состояния.***

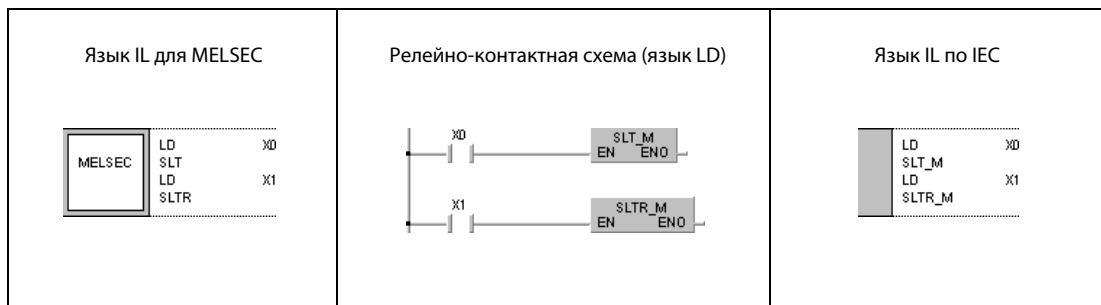
*Дополнительная информация на тему фиксации состояния имеется в руководствах по центральным процессорам и руководстве по работе со средой GX IEC Developer.*

*При выполнении команды SLT сильно увеличивается время выполнения цикла программы в центральном процессоре. Время обработки команды SLT указано в руководствах по соответствующим центральным процессорам. Его следует учитывать при настройке сторожевых таймеров (только серия "A").*

**Пример**

SLT/SLTR

В следующей программе при включенном входе X0 выполняется команда SLT. При включенном входе X1 команда SLTR сбрасывает команду SLT.



7.10.5 STRA, STRAR

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	

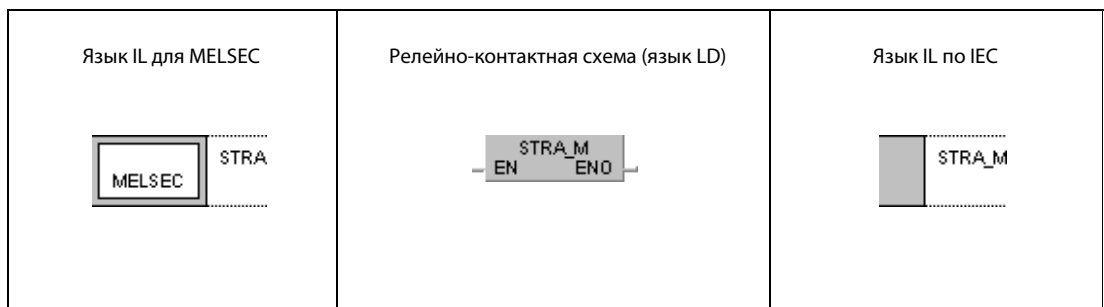
Операнды MELSEC A

Операнды															Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки						
битовые					словные (16 бит)					Константы		Указатели		Уровень											
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V	K	H (16#)	P	I	N
																						1			M9010 M9011

Операнды MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., польз.)		Регистр файлов	MELSECNET/10 непос. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы	Иные			
битовые	словные		битовые	словные							
—	—	—	—	—	—	—	—	—	—		1

GX IEC Developer



Переменные

Операнд	Значение	Тип данных
—	—	—



<b>Принцип действия</b>	<b>Установка и сброс выборочной трассировки</b>
	<b>STRA    Установка выборочной трассировки (семплирования)</b>  Функция выборочной трассировки контролирует данные и состояния выбранных операндов через определенные интервалы и сохраняет накопленные данные этих операндов в отдельной области памяти. Выбор операндов и интервала опроса осуществляется с помощью параметров.  <b>STRAR    Сброс выборочной трассировки (семплирования)</b>  Команда STRAR стирает данные в программном файле выборочной трассировки и сбрасывает команду STRA и специальный маркер M9043 (серия "A") или маркеры SM801–SM805 (серии "Q"/"System Q").  Повторное выполнение команды STRA возможно лишь после выполнения команды STRAR.
<b>ПРИМЕЧАНИЯ</b>	<b><i>Пожалуйста, проверьте, поддерживает ли ваша версия среды GX IEC Developer функцию выборочной трассировки.</i></b> <i>Дополнительная информация на тему выборочной трассировки имеется в руководствах по центральным процессорам и в руководстве по работе в среде GX IEC Developer.</i> <i>Во время обращения к ROM выполнение команд STRA и STRAR не возможно (только серия "A").</i>

**Пример**

STRA/STRAR

В следующей программе при включенном входе X0 выполняется команда STRA. Команда STRA сбрасывается в результате выполнения команды STRAR. Команда STRAR выполняется при включенном входе X1.



**7.10.6 PTRA, PTRAR, PTRAEXE, PTRAEXEP**

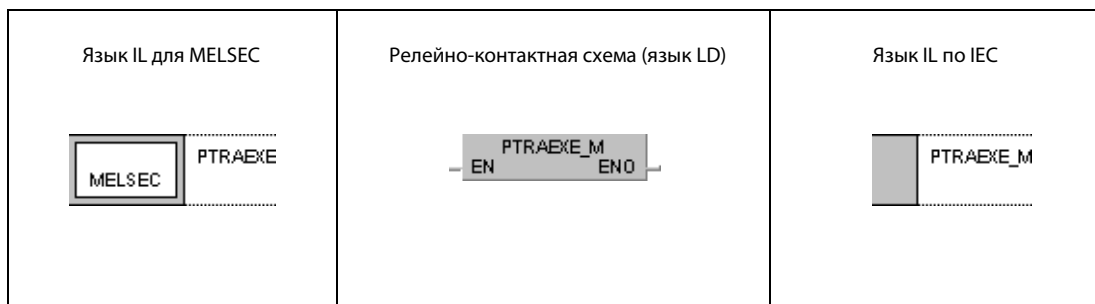
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

**Операнды MELSEC Q**

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистр файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы	Иные			
битовые	словные		битовые	словные							
—	—	—	—	—	—	—	—	—	—	—	1

**GX IEC Developer**



**Переменные**

Операнд	Значение	Тип данных
—	—	—

<b>Принцип действия</b>	<b>Выполнение, установка и сброс трассировки программы</b>
	<b>PTRA                      Установка трассировки (контроля) программы</b>
	Функция трассировки программы контролирует данные и состояния установленных программой операндов в течение определенного интервала и сохраняет накопленные данные контролируемых программ в отдельной области памяти.  Команда PTRA позволяет контролировать программы в течение заданного количества циклов и сохранять данные в отдельной области промежуточной памяти центрального процессора для последующей функции трассировки. Сама трассировка программы запускается командой PTRAEHE. Предварительным условием для сохранения данных является установленное состояние (1) маркеров SM810...SM812.  При выполнении команды PTRA устанавливается маркер SM813. После выполнения заданного количества циклов опроса данные сохраняются для дальнейшей обработки и трассировка программы прекращается.  Если во время трассировки программы сбрасывается маркер SM811, процесс трассировки прерывается.  После завершения выполнения команды PTRA устанавливается маркер SM815.  Перед повторным выполнением команды PTRA должна быть выполнена команда PTRAR.  Результаты трассировки можно контролировать с периферийного устройства.
	<b>PTRAR                      Сброс трассировки (контроля) программы</b>
Команд PTRAR стирает данные в программном файле трассировки программы и сбрасывает команду PTRA и маркеры SM811...SM815.  Повторное выполнение команды PTRA возможно лишь после выполнения команды PTRAR.	
	<b>PTRAEHE                      Выполнение трассировки (контроля) программы</b>
	Команда PTRAEHE запускает трассировку программы.  Если во время трассировки программы сбрасывается маркер SM811, процесс трассировки прерывается.  Если условие выполнения команды PTRAEHE не установлено, трассировка не выполняется.
<b>ПРИМЕЧАНИЯ</b>	<b><i>Пожалуйста, проверьте, поддерживает ли ваша версия среды GX IEC Developer функцию трассировки программы.</i></b> <i>Дополнительная информация на тему трассировки программы имеется в руководствах по центральным процессорам и в руководстве по работе в среде GX IEC Developer.</i>

## 7.11 Команды для обработки строковых величин

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе IEC
Преобразование 16-битных или 32-битных двоичных данных в десятичные числа в кодировке ASCII	BINDA	BINDA_MD
		BINDA_K_MD
		BINDA_S_MD
	BINDAP	BINDA_P_MD
		BINDA_K_P_MD
		BINDA_P_S_MD
	DBINDA	DBINDA_MD
		DBINDA_K_MD
		DBINDA_S_MD
	DBINDAP	DBINDA_P_MD
		DBINDA_K_P_MD
		DBINDA_P_S_MD
Преобразование 16-битных или 32-битных двоичных данных в шестнадцатеричные числа в кодировке ASCII	BINHA	BINHA_MD
		BINHA_K_MD
		BINHA_S_MD
	BINHAP	BINHA_P_MD
		BINHA_K_P_MD
		BINHA_P_S_MD
	DBINHA	DBINHA_MD
		DBINHA_K_MD
		DBINHA_S_MD
	DBINHAP	DBINHA_P_MD
		DBINHA_K_P_MD
		DBINHA_P_S_MD
Преобразование 4/8-значных двоично-десятичных данных в ASCII-код	BCDDA	BCDDA_MD
		BCDDA_K_MD
		BCDDA_S_MD
	BCDDAP	BCDDA_P_MD
		BCDDA_K_P_MD
		BCDDA_P_S_MD
	DBCDDA	DBCDDA_MD
		DBCDDA_K_MD
		DBCDDA_S_MD
	DBCDDAP	DBCDDA_P_MD
		DBCDDA_K_P_MD
		DBCDDA_P_S_MD

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе IEC
Преобразование десятичных данных в кодировке ASCII в 16-битные или 32-битные двоичные данные	DABIN	DABIN_MD
		DABIN_S_MD
	DABINP	DABIN_P_MD
		DABIN_P_S_MD
	DDABIN	DDABIN_MD
		DDABIN_S_MD
DDABINP	DDABIN_P_MD	
	DDABIN_P_S_MD	
Преобразование шестнадцатеричных данных в кодировке ASCII в 16-битные или 32-битные двоичные данные	HABIN	HABIN_MD
		HABIN_S_MD
	HABINP	HABIN_P_MD
		HABIN_P_S_MD
	DHABIN	DHABIN_MD
		DHABIN_S_MD
DHABINP	DHABIN_P_MD	
	DHABIN_P_S_MD	
Преобразование десятичных данных в кодировке ASCII в 4-/8-значные двоично-десятичные данные	DABCD	DABCD_MD
		DABCD_S_MD
	DABCDP	DABCD_P_MD
		DABCD_P_S_MD
	DDABCD	DDABCD_MD
		DDABCD_S_MD
DDABCDP	DDABCD_P_MD	
	DDABCD_P_S_MD	
Считывание данных комментариев	COMRD	COMRD_MD
		COMRD_S_MD
	COMRDP	COMRD_P_MD
		COMRD_P_S_MD
Определение длины строковых величин	LEN	LEN_E
		LEN_MD
		LEN_S_MD
	LENP	LEN_P_S_MD
Преобразование 16-битных или 32-битных двоичных данных в строковые величины	STR	STR_MD
		STR_K_MD
		STR_S_MD
	STRP	STR_P_MD
		STR_K_P_MD
		STR_P_S_MD
	DSTR	DSTR_MD
		DSTR_K_MD
		DSTR_S_MD
	DSTRP	DSTR_P_MD
DSTR_K_P_MD		
DSTR_P_S_MD		

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе IEC
Преобразование строковых величин в 16-битные или 32-битные двоичные данные	VAL	VAL_MD
		VAL_S_MD
	VALP	VAL_P_MD
		VAL_P_S_MD
	DVAL	DVAL_MD
		DVAL_S_MD
DVALP	DVAL_P_MD	
	DVAL_P_S_MD	
Преобразование чисел с плавающей запятой в строковые величины	ESTR	ESTR_M
	ESTRP	ESTRP_M
Преобразование строковых величин в десятичные числа с плавающей запятой	EVAL	EVAL_M
	EVALP	EVALP_M
Преобразование буквенно-числовых строковых величин в ASCII-код	ASC	ASC_MD
		ASC_K_MD
		ASC_S_MD
	ASCP	ASC_P_MD
		ASC_P_S_MD
		ASC_K_P_MD
Преобразование шестнадцатеричных значений ASCII в двоичные значения	HEX	HEX_MD
		HEX_K_MD
		HEX_S_MD
	HEXP	HEX_P_MD
		HEX_K_P_MD
		HEX_P_S_MD
Выделение фрагмента (правой части) строковой величины	RIGHT	RIGHT_M
	RIGHTP	RIGHTP_M
Выделение фрагмента (левой части) строковой величины	LEFT	LEFT_M
	LEFTP	LEFTP_M
Выделение произвольных фрагментов строковой величины	MIDR	MIDR_M
	MIDRP	MIDRP_M
	MIDW	MIDW_M
	MIDWP	MIDWP_M
Поиск строковых величин	INSTR	INSTR_M
	INSTRP	INSTRP_M
Преобразование числа с плавающей запятой в двоично-десятичный формат	EMOD	EMOD_M
	EMODP	EMODP_M
Преобразование числа с плавающей запятой в десятичный формат	EREXP	EREXP_M
	EREXPP	EREXPP_M

7.11.1 BINDA, BINDAP, DBINDA, DBINDAP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

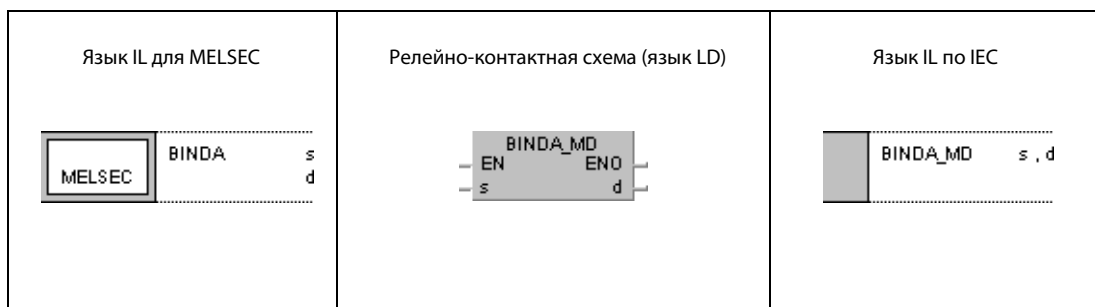
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

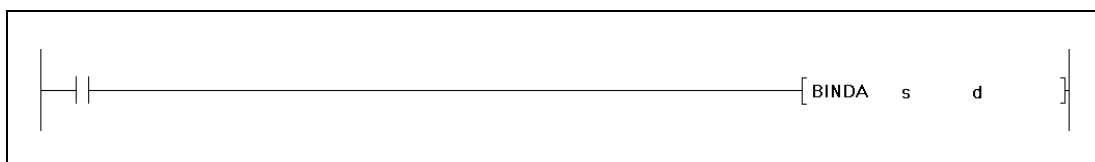
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	—	—	3	
d	—	●	●	—	—	—	—	—	—		

GX IEC Developer



GX Developer



Переменные

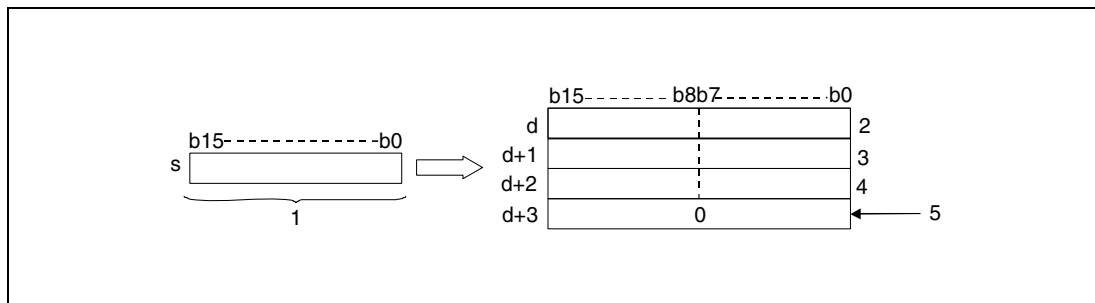
Операнд	Значение	Тип данных	
		MELSEC	IEC
s	Двоичные данные, конвертируемые в формат ASCII.	BIN, 16/32 бита	ANY16/32
d	Первый адрес операнда, в котором сохраняется результат конвертации.	строковая величина	массив [1..4]/[1..6] данных типа ANY16



**Принцип действия**

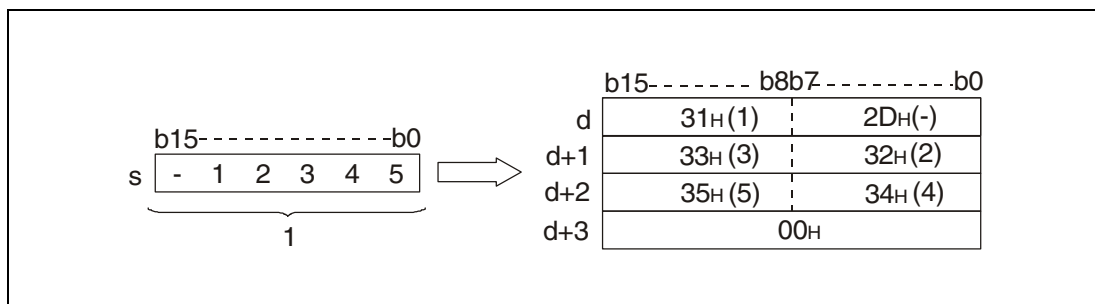
**Преобразование 16/32-битных двоичных данных в десятичные числа в кодировке ASCII**  
**BINDA Преобразование 16-битных двоичных данных**

Команда BINDA преобразует 16-битное двоичное число, указанное в s, в десятичное число в кодировке ASCII и сохраняет его в регистрах от d (Array\_d[1]) до d + 3 (Array\_d[4]).



- <sup>1</sup> 16-битные двоичные данные
- <sup>2</sup> разряд десятков тысяч в кодировке ASCII/арифметический знак
- <sup>3</sup> разряд сотен в кодировке ASCII/разряд тысяч в кодировке ASCII
- <sup>4</sup> разряд единиц в кодировке ASCII/разряд десятков в кодировке ASCII
- <sup>5</sup> при не установленном маркере SM701

Указанное в s значение сохраняется в виде десятичного числа в кодировке ASCII в регистрах от d (Array\_d[1]) до d + 3 (Array\_d[4]).



- <sup>1</sup> двоичное число

16-битное двоичное число, указанное в s, может находиться в диапазоне между -32768 и 32767.

Результаты конвертации сохраняются начиная с d следующим образом:

Если 16-битное двоичное число положительное, арифметический знак записывается в виде кода "20н".

В случае отрицательного двоичного числа арифметический знак записывается в виде кода "2Dн".

Записанный арифметический знак "20н" заменяет собой предшествующие нули.

В числе 00325 нули в разрядах десятков тысяч и тысяч заменяются кодами "20н". Таким образом, сохраняются только действительно необходимые разряды.

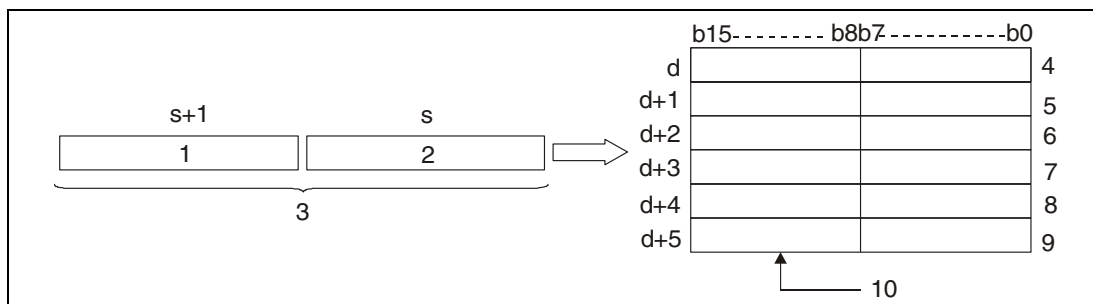
Сохранение данных в операнде, указанном в d + 3 (Array\_d[4]), зависит от состояния маркера SM701.

Если маркер не установлен, в область d + 3 (Array\_d[4]) записывается нулевой код "00н".

Если маркер установлен, значение в d + 3 (Array\_d[4]) остается без изменений.

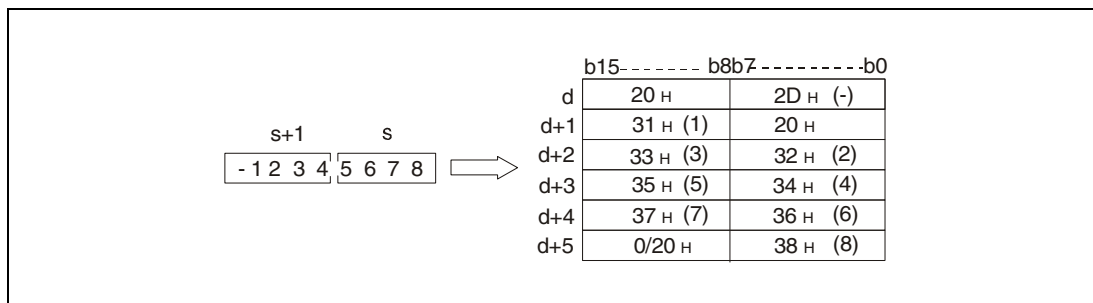
**DBINDA Преобразование 32-битных двоичных данных**

Команда DBINDA преобразует 32-битные двоичные данные, указанные в s и s + 1, в десятичное число в кодировке ASCII и сохраняет его в операндах от d (Array\_d[1]) до d + 5 (Array\_d[6]).



- <sup>1</sup> старшие 16 битов
- <sup>2</sup> младшие 16 битов
- <sup>3</sup> 32-битные двоичные данные
- <sup>4</sup> арифметический знак/разряд миллиардов в кодировке ASCII
- <sup>5</sup> разряд десятков миллионов/разряд сотен миллионов в кодировке ASCII
- <sup>6</sup> разряд сотен тысяч/разряд миллионов в кодировке ASCII
- <sup>7</sup> разряд тысяч/разряд десятков тысяч в кодировке ASCII
- <sup>8</sup> разряд десятков/разряд сотен в кодировке ASCII
- <sup>9</sup> 0 или 20н/разряд единиц в кодировке ASCII
- <sup>10</sup> при не установленном маркере SM701 (0)/при установленном маркере SM701 (20н)

Указанное в s и s + 1 значение сохраняется в виде десятичного числа в кодировке ASCII в операндах от d (Array\_d[1]) до d + 5 (Array\_d[6]).



32-битное двоичное число, указанное в s, может находиться в диапазоне между -2147483648 и 2147483647.

Результаты конвертации сохраняются в операндах от d (Array\_d[1]) до d + 5 (Array\_d[6]) следующим образом:

Если двоичное число положительное, арифметический знак записывается в виде кода "20н".  
 В случае отрицательного двоичного числа арифметический знак записывается в виде кода "2Dн".

Записанный арифметический знак "20н" заменяет собой предшествующие нули.  
 В числе 0012034560 нули в разрядах миллиардов и сотен миллионов заменяются кодами "20н".  
 Таким образом, сохраняются только действительно необходимые разряды.

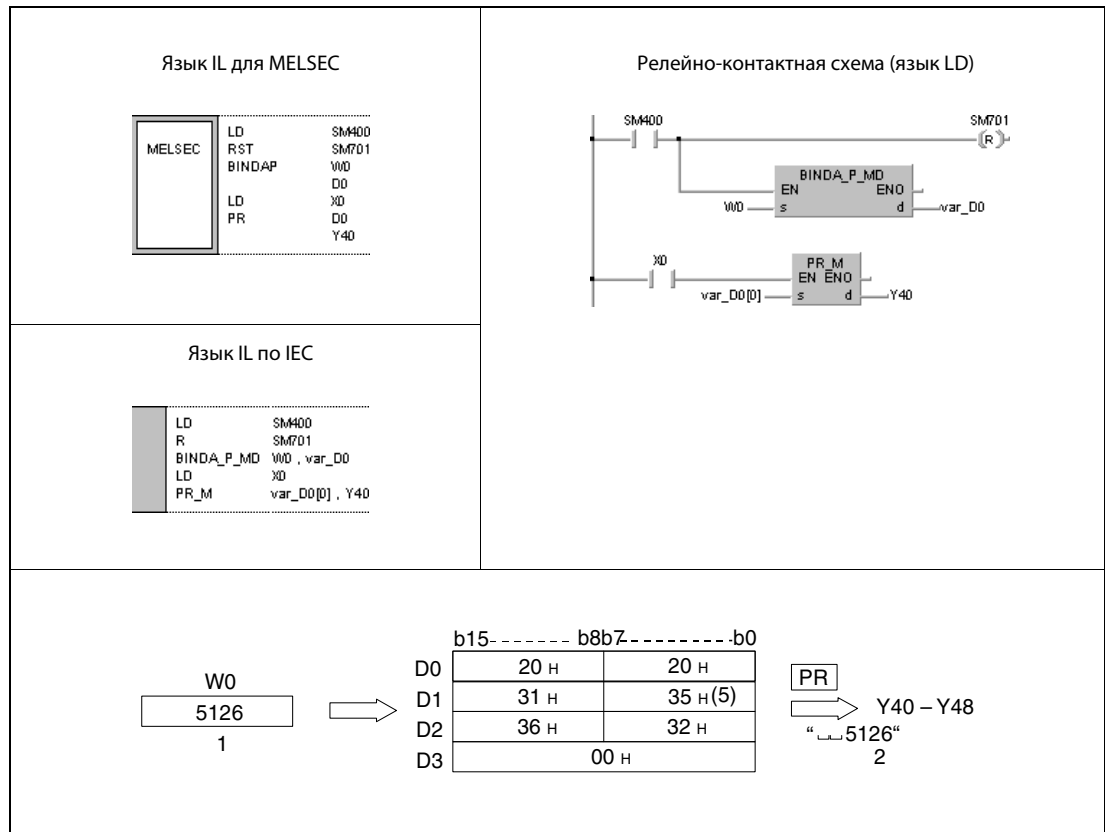
Сохранение данных в старших 8 битах операнда d + 5 (Array\_d[6]) зависит от состояния маркера SM701 следующим образом:

Если маркер не установлен, в область d + 5 (Array\_d[6]) записывается нулевой код "00н".

Если маркер установлен, в область d + 5 (Array\_d[6]) записывается знак пробела (20н).

**Пример 1** BINDAP

Следующая программа при положительном фронте сигнала SM400 с помощью команды BINDAP преобразует значение 16-битных двоичных данных W0 в десятичное значение в кодировке ASCII. Знаки выводятся командой PR на выходы с Y40 по Y48.

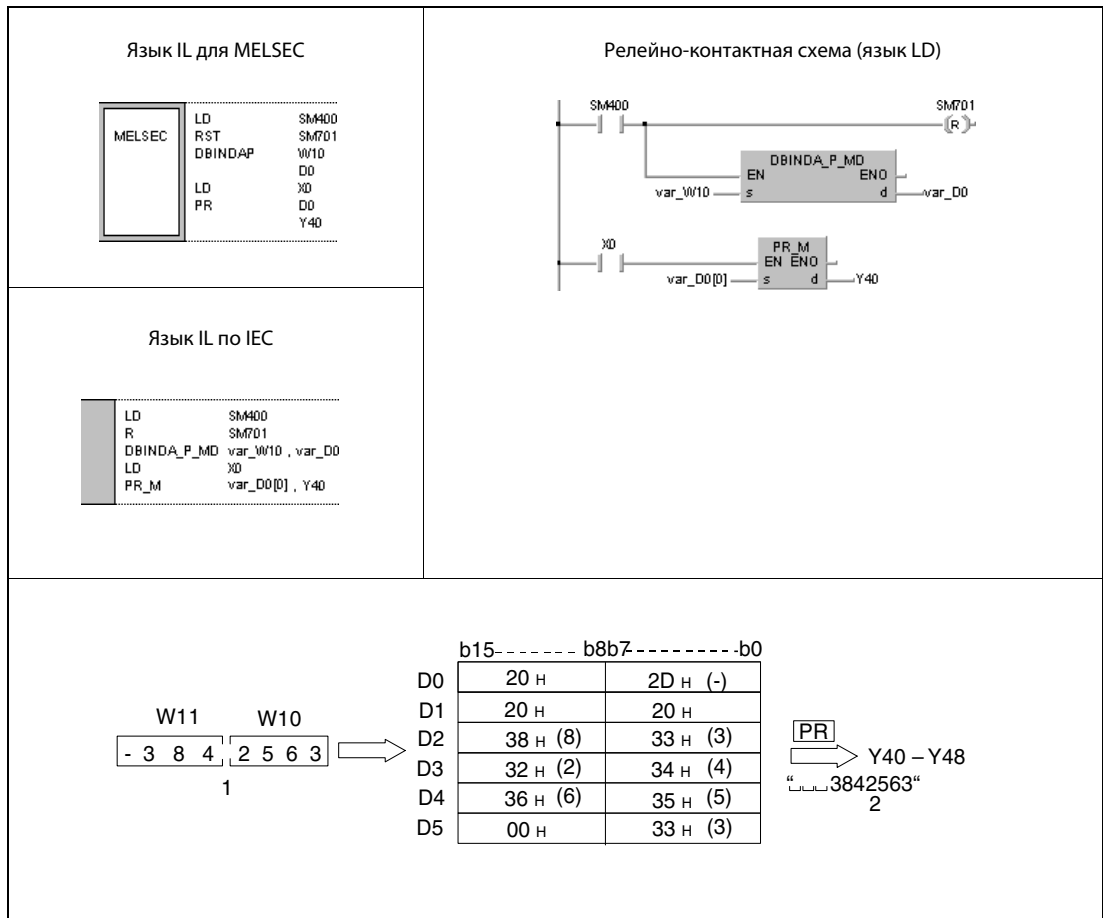


<sup>1</sup> двоичное число

<sup>2</sup> вывод

**Пример 2** DBINDAP

Следующая программа при положительном фронте сигнала SM400 с помощью команды DBINDAP преобразует значение 32-битных двоичных данных W10 и W11 в десятичное значение в кодировке ASCII. Знаки выводятся командой PR на выходы с Y40 по Y48.



<sup>1</sup> вывод

<sup>2</sup> двоичное число

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.11.2 BINHA, BINHAP, DBINHA, DBINHAP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

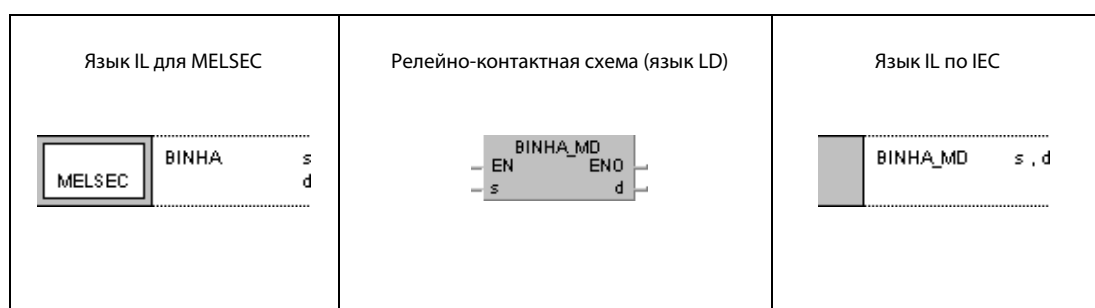
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

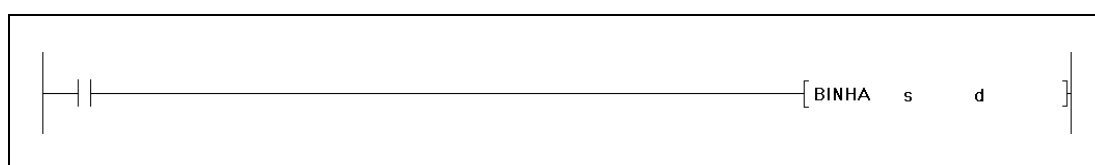
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	●	—	—	3
d	—	●	●	—	—	—	—	—	—	—	

**GX IEC Developer**



**GX Developer**



**Переменные**

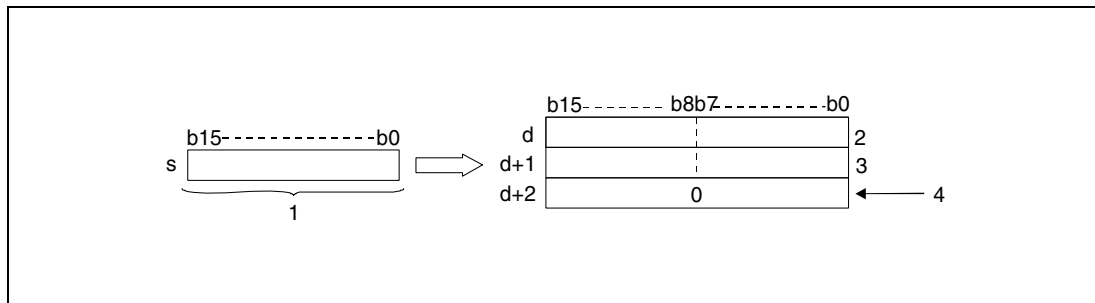
Операнд	Значение	Тип данных	
		MELSEC	IEC
s	Двоичные данные, конвертируемые в формат ASCII.	BIN, 16/32 бита	ANY16/32
d	Первый адрес операнда, в котором сохраняется результат конвертации.	строковая величина	массив [1..3]/[1..5] данных типа ANY16

**Принцип действия**

**Преобразование 16/32-битных двоичных данных в шестнадцатеричные числа в кодировке ASCII**

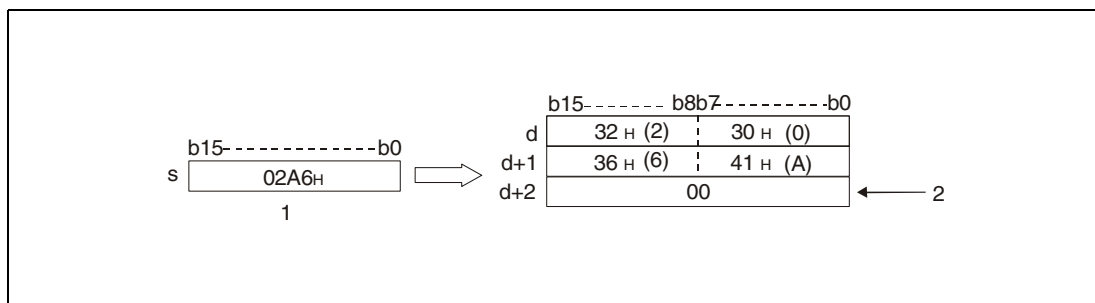
**BINHA Преобразование 16-битных двоичных данных**

Команда BINHA преобразует 16-битные двоичные данные, указанные в s, в шестнадцатеричное число в кодировке ASCII и сохраняет его в операндах от d (Array\_d[1]) до d + 2 (Array\_d[3]).



- <sup>1</sup> 16-битные двоичные данные
- <sup>2</sup> ASCII-код 3-го разряда/ASCII-код 4-го разряда
- <sup>3</sup> ASCII-код 1-го разряда/ASCII-код 2-го разряда
- <sup>4</sup> при не установленном маркере SM701

Указанное в s значение сохраняется в кодировке ASCII в операндах от d (Array\_d[1]) до d + 2 (Array\_d[3]).



- <sup>1</sup> 16-битные двоичные данные
- <sup>2</sup> при не установленном маркере SM701

16-битные двоичные данные, указанные в s, могут быть в диапазоне между 0н и FFFFн.

Результат конвертации сохраняется в виде 4-значного шестнадцатеричного значения в операндах от d (Array\_d[1]) до d + 2 (Array\_d[3]).

Если значение разряда равно нулю, этот разряд обрабатывается как ноль (т. е. нулевые разряды не подавляются).

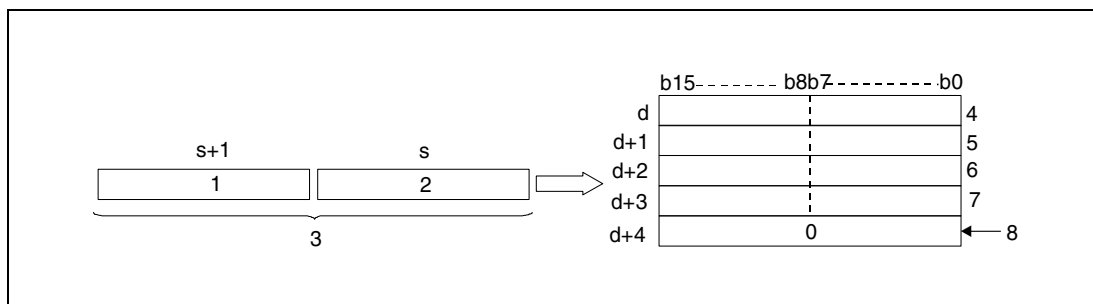
Сохранение данных в операнде, указанном в d + 2 (Array\_d[3]), зависит от состояния маркера SM701 следующим образом:

Если маркер не установлен, в область d + 2 (Array\_d[3]) записывается нулевой код "00н".

Если маркер установлен, значение в d + 2 (Array\_d[3]) остается без изменений.

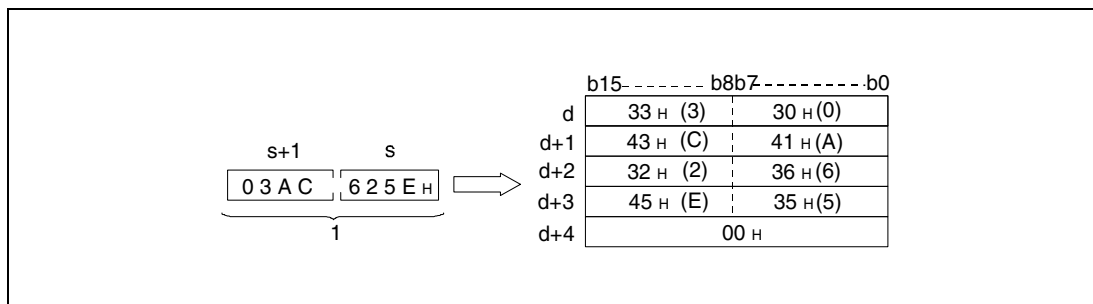
**DBINHA Преобразование 32-битных двоичных данных**

32-битные двоичные данные, указанные в s и s + 1, преобразуются в шестнадцатеричный код ASCII и сохраняются в операндах от d (Array\_d[1]) до d + 4 (Array\_d[5]).



- <sup>1</sup> старшие 8 битов
- <sup>2</sup> младшие 8 битов
- <sup>3</sup> 32-битные двоичные данные
- <sup>4</sup> ASCII-код 7-го разряда/ASCII-код 8-го разряда
- <sup>5</sup> ASCII-код 5-го разряда/ASCII-код 6-го разряда
- <sup>6</sup> ASCII-код 3-го разряда/ASCII-код 4-го разряда
- <sup>7</sup> ASCII-код 1-го разряда/ASCII-код 2-го разряда
- <sup>8</sup> при не установленном маркере SM701

Указанное в s и s + 1 значение "03AC625Eн" сохраняется, начиная с d, следующим образом:



<sup>1</sup> BIN, 32-битные данные

32-битное двоичное значение, указанное в s и s + 1, может находиться в диапазоне между 0H и FFFFFFFFH.

Результат конвертации сохраняется в виде 8-значного шестнадцатеричного значения в операндах от d (Array\_d[1]) до d + 4 (Array\_d[5]).

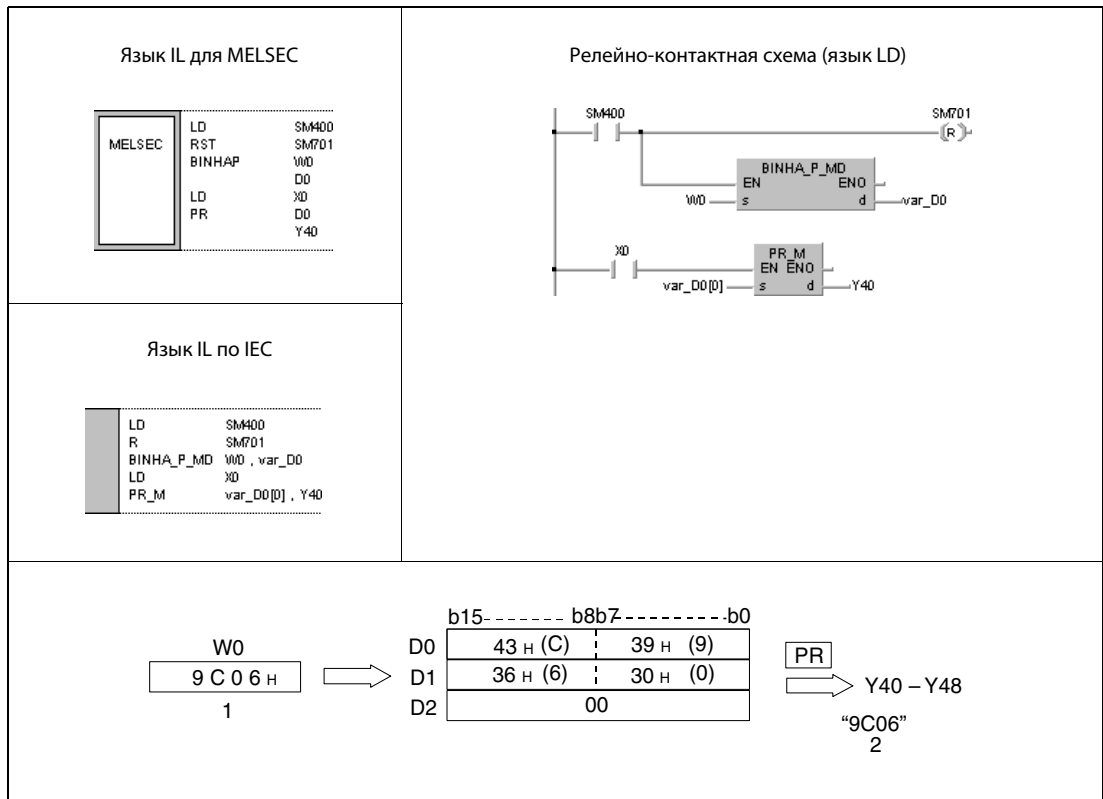
Если значение разряда равно нулю, этот разряд обрабатывается как ноль (т. е. предшествующие нули не подавляются).

Сохранение данных в операнде, указанном в d + 4 (Array\_d[5]), зависит от состояния маркера SM701 следующим образом:

Если маркер не установлен, в область d + 4 (Array\_d[5]) записывается код "00н" (ноль).  
 Если маркер установлен, значение в d + 4 (Array\_d[5]) остается без изменений.

**Пример 1** BINHAP

Следующая программа при положительном фронте сигнала SM400 с помощью команды BINHAP преобразует значение 16-битных двоичных данных W0 в шестнадцатеричное значение в кодировке ASCII. Знаки выводятся командой PR на выходы с Y40 по Y48.



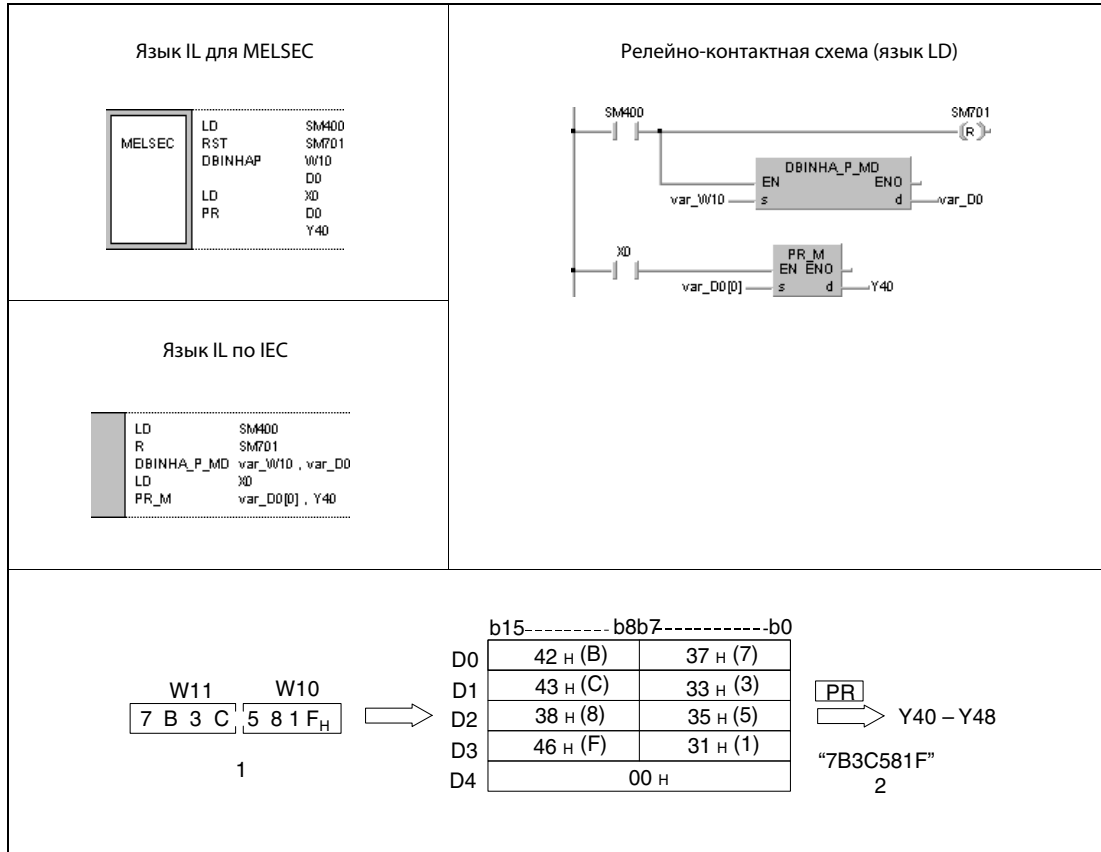
<sup>1</sup> вывод

<sup>2</sup> двоичное число



**Пример 2** DBINHAP

Следующая программа при положительном фронте сигнала SM400 с помощью команды DBINHAP преобразует значение 32-битных двоичных данных W10 и W11 в шестнадцатеричное значение в кодировке ASCII. Знаки выводятся командой PR на выходы с Y40 по Y48.



<sup>1</sup> вывод

<sup>2</sup> двоичное число

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

7.11.3 BCDDA, BCDDAP, DBCDDA, DBCDDAP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

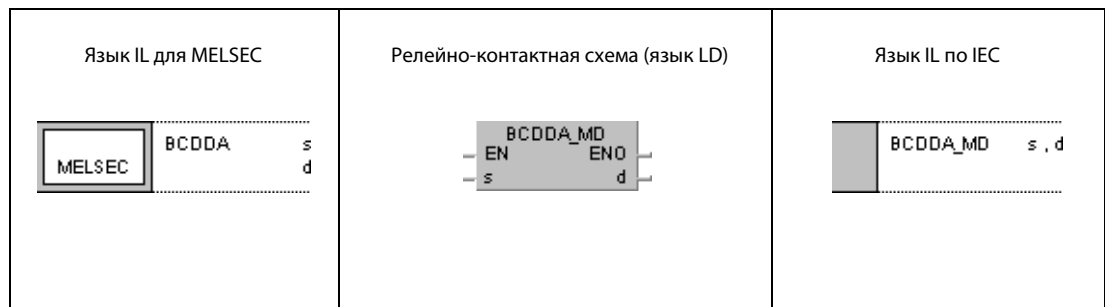
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

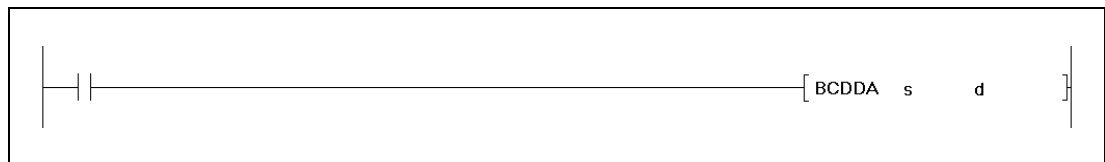
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□NG□	Индексные регистры Zn	Константы \$			Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	—	SM0	3	
d	—	●	●	—	—	—	—	—			

GX IEC Developer



GX Developer



Переменные

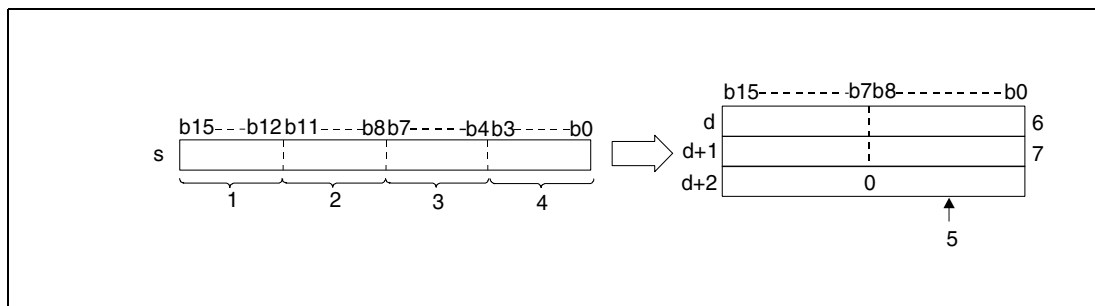
Операнд	Значение	Тип данных	
		MELSEC	IEC
s	Двоично-десятичные данные, конвертируемые в формат ASCII.	слово	ANY16/32
d	Первый адрес, начиная с которого сохраняется результат конвертации.	строковая величина	массив [1..3]/ [1..5] данных типа ANY16

**Принцип действия**

**Преобразование 4/8-значных двоично-десятичных данных в ASCII-код**

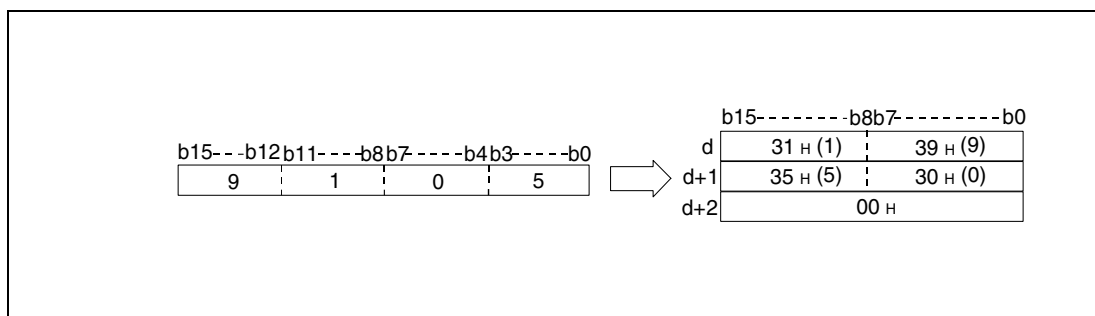
**BCDDA Преобразование 4-значных двоично-десятичных данных**

Команда BCDDA преобразует 4-значные двоично-десятичные данные, указанные в s, в формат ASCII и сохраняет результат в операндах от d (Array\_d[1]) до d + 2 (Array\_d[3]).



- 1 разряд тысяч
- 2 разряд сотен
- 3 разряд десятков
- 4 разряд единиц
- 5 при не установленном маркере SM701
- 6 ASCII-код 3-го разряда/ASCII-код 4-го разряда
- 7 ASCII-код 1-го разряда/ASCII-код 2-го разряда

Указанное в s значение 9105 сохраняется в d следующим образом:



Указанное в s двоично-десятичное значение может быть в диапазоне между 0 и 9999.

Результат конвертации сохраняется в операндах от d (Array\_d[1]) до d + 2 (Array\_d[3]).

Если значение разряда равно нулю, то этот разряд сохраняется в виде кода нуля "30н" (т. е. нулевые разряды не подавляются).

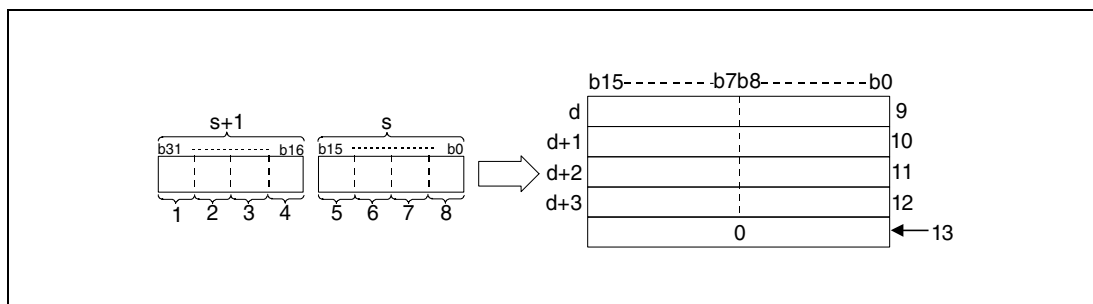
Сохранение данных в операнде, указанном в d + 2 (Array\_d[3]), зависит от состояния маркера SM701 следующим образом:

Если маркер не установлен, в область d + 2 (Array\_d[3]) записывается код "00н" (ноль).

Если маркер установлен, значение в d + 2 (Array\_d[3]) остается без изменений.

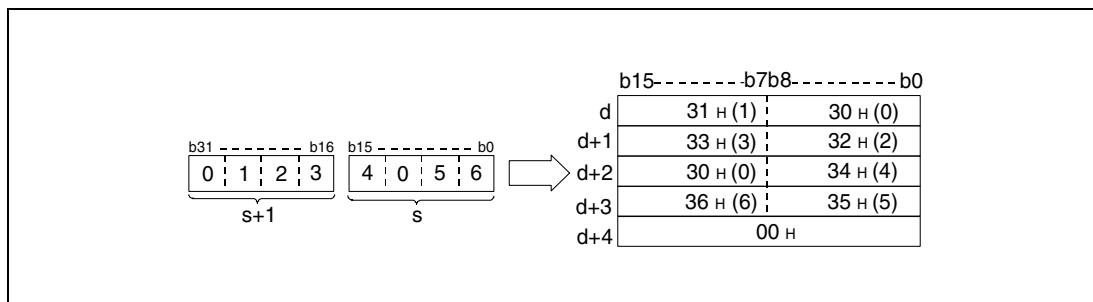
**DBCDDA      Преобразование 8-значных двоично-десятичных данных**

Команда DBCDDA преобразует 8-значные двоично-десятичные данные, указанные в  $s$  и  $s + 1$ , в формат ASCII и сохраняет результат в операндах от  $d$  (Array\_d[1]) до  $d + 4$  (Array\_d[5]).



- <sup>1</sup> разряд десятков миллионов
- <sup>2</sup> разряд миллионов
- <sup>3</sup> разряд сотен тысяч
- <sup>4</sup> разряд десятков тысяч
- <sup>5</sup> разряд тысяч
- <sup>6</sup> разряд сотен
- <sup>7</sup> разряд десятков
- <sup>8</sup> разряд единиц
- <sup>9</sup> ASCII-код 7-го разряда/ASCII-код 8-го разряда
- <sup>10</sup> ASCII-код 5-го разряда/ASCII-код 6-го разряда
- <sup>11</sup> ASCII-код 3-го разряда/ASCII-код 4-го разряда
- <sup>12</sup> ASCII-код 1-го разряда/ASCII-код 2-го разряда
- <sup>13</sup> при не установленном маркере SM701

Указанное в  $s$  и  $s + 1$  значение 01234056 сохраняется, начиная с  $d$ , следующим образом:



Двоично-десятичное значение, указанное в  $s$  и  $s + 1$ , может быть в диапазоне между 0 и 99999999.

Результат конвертации сохраняется в операндах от  $d$  (Array\_d[1]) до  $d + 4$  (Array\_d[5]).

Если значение разряда равно нулю, тот этот разряд сохраняется в виде кода нуля "30н" (т. е. нулевые разряды не подавляются).

Сохранение данных в операнде, указанном в  $d + 4$  (Array\_d[5]), зависит от состояния маркера SM701.

Если маркер не установлен, в область  $d + 4$  (Array\_d[5]) записывается нулевой код "00н".

Если маркер установлен, значение в  $d + 4$  (Array\_d[5]) остается без изменений.

**Источники ошибок**

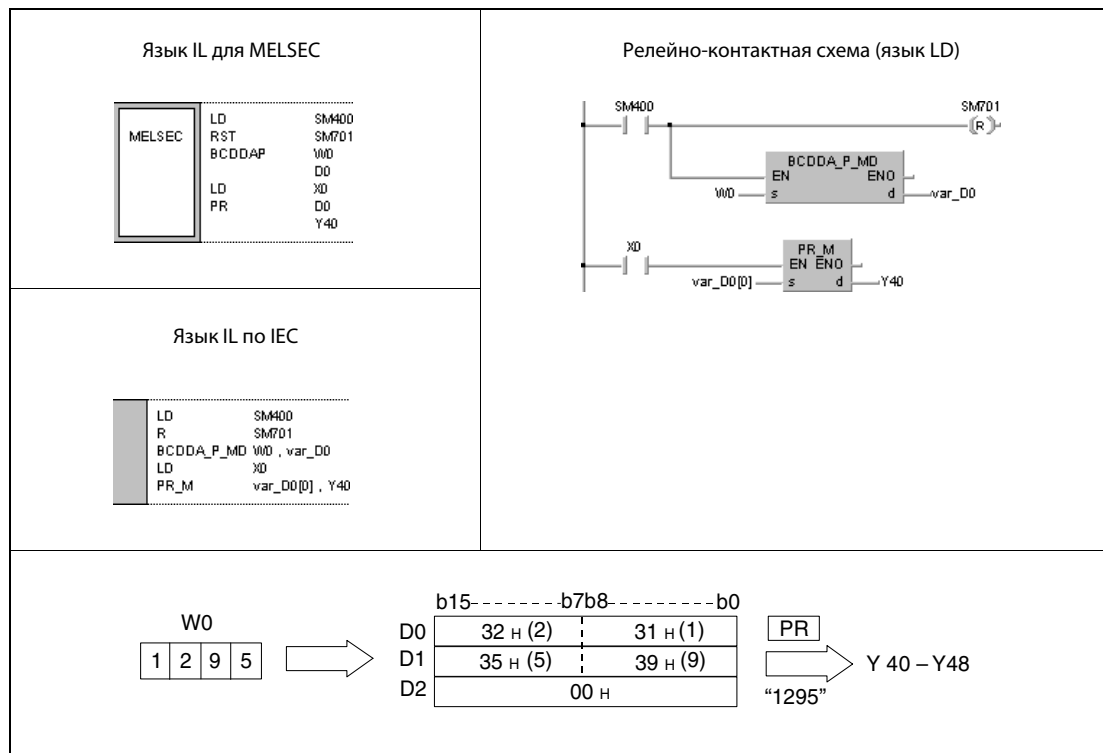
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Во время обработки команды BCDDA двоично-десятичные данные в s находятся вне диапазона от 0 до 9999 (код ошибки 4100).
- Во время обработки команды DBCDDA двоично-десятичные данные в s находятся вне диапазона от 0 до 99999999 (код ошибки 4100)

**Пример 1**

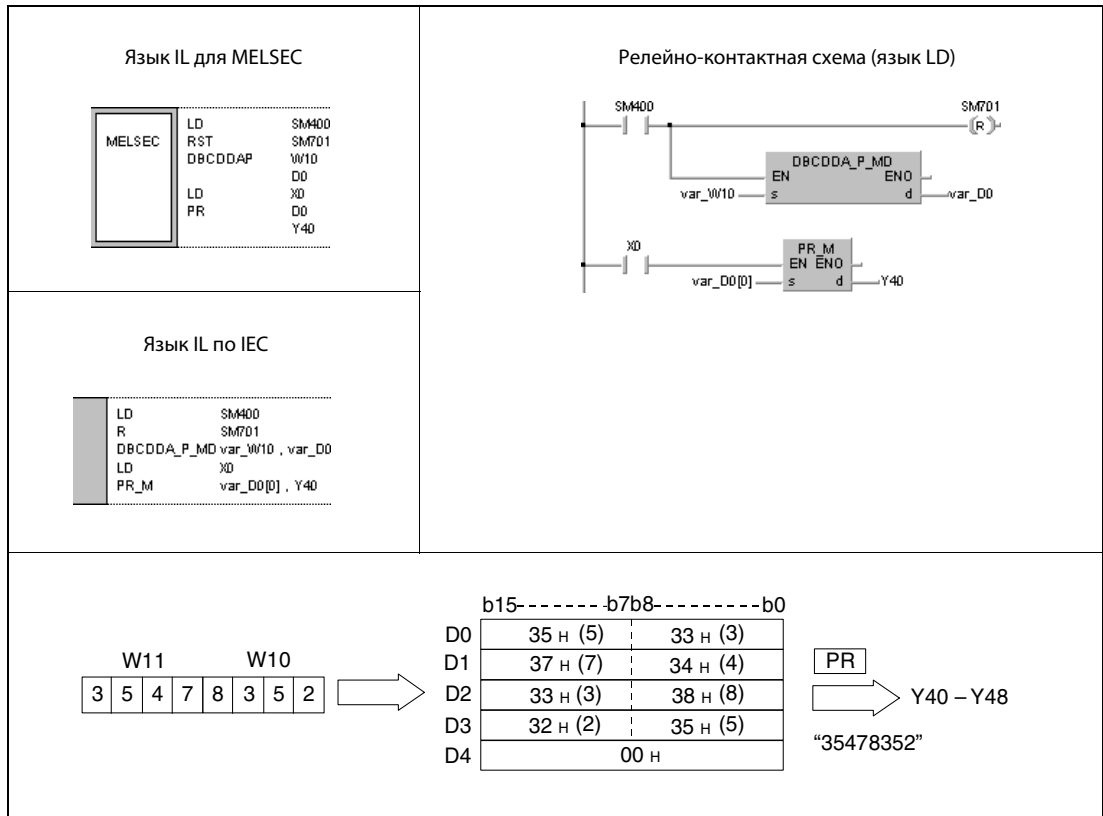
**BCDDAP**

Следующая программа при положительном фронте сигнала SM400 с помощью команды BCDDAP преобразует 4-значное двоично-десятичное значение W0 в десятичное значение в кодировке ASCII. Знаки выводятся командой PR на выходы с Y40 по Y48.



**Пример 2** DBCDDAP

Следующая программа при положительном фронте сигнала SM400 с помощью команды DBCDDAP преобразует 8-значное двоично-десятичное значение W10 и W11 в десятичное значение в кодировке ASCII. Знаки выводятся командой PR на выходы с Y40 по Y48.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.11.4 DABIN, DABINP, DDABIN, DDABINP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

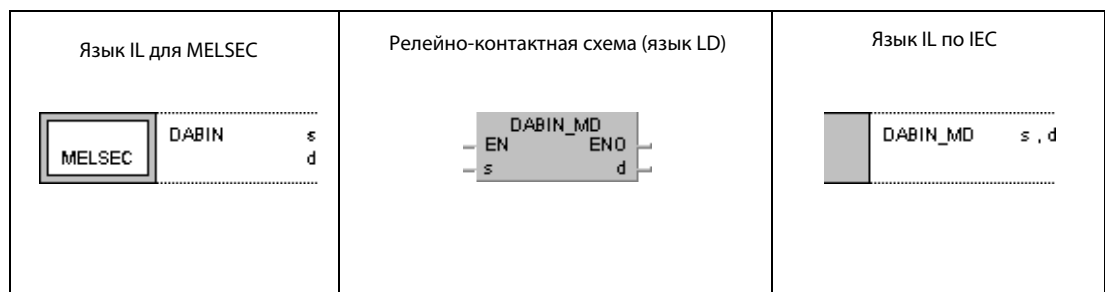
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

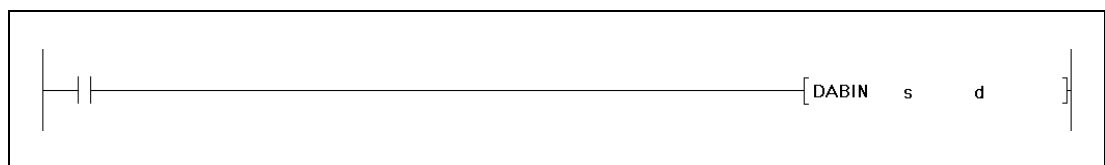
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□NG□	Индексные регистры Zn	Константы \$			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

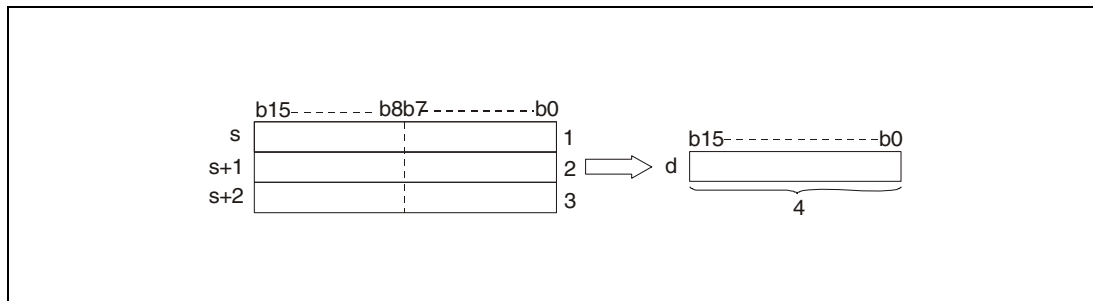
Операнд	Значение	Тип данных	
		MELSEC	IEC
s	Область памяти, в которой сохранены конвертируемые данные в формате ASCII.	строковая величина	массив [1..3]/ [1..6] данных типа ANY16
d	Область памяти, в которой сохраняется результат конвертации.	BIN, 16/32 бита	ANY16/32

**Принцип действия**

**Преобразование десятичных данных в кодировке ASCII в 16-битные и 32-битные двоичные данные**

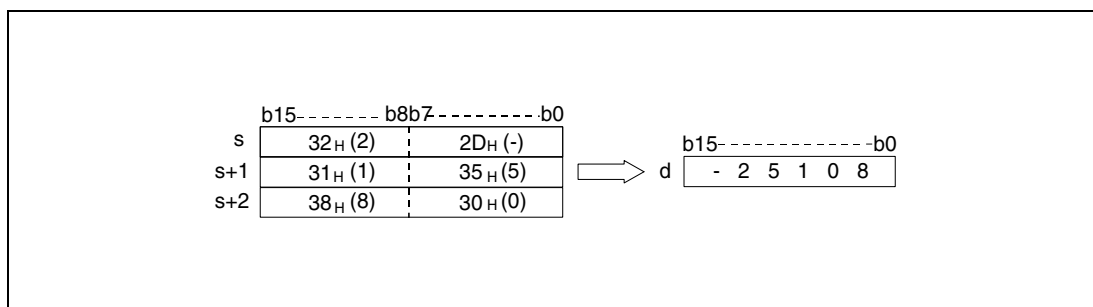
**DABIN Преобразование в 16-битные двоичные данные**

Команда DABIN преобразует десятичные данные в кодировке ASCII, указанные в операндах от s (Array\_s[1]) до s + 2 (Array\_s[3]), в 16-битный двоичный формат и сохраняет результат в d.



- <sup>1</sup> ASCII-код разряда десятков тысяч/арифметический знак
- <sup>2</sup> ASCII-код разряда сотен/ASCII-код разряда тысяч
- <sup>3</sup> ASCII-код разряда единиц/ASCII-код разряда десятков
- <sup>4</sup> 16-битные двоичные данные

Значение, указанное в операндах от s (Array\_s[1]) до s + 2 (Array\_s[3]), сохраняется в d как -25018<sub>BIN</sub> следующим образом:



Значение в кодировке ASCII, указанное в операндах от s (Array\_s[1]) до s + 2 (Array\_s[3]), может находиться в диапазоне между -32768 и 32767.

Если двоичное значение положительное, арифметический знак сохраняется в виде кода "20н".

При отрицательном значении арифметический знак сохраняется в виде кода "2Dн".

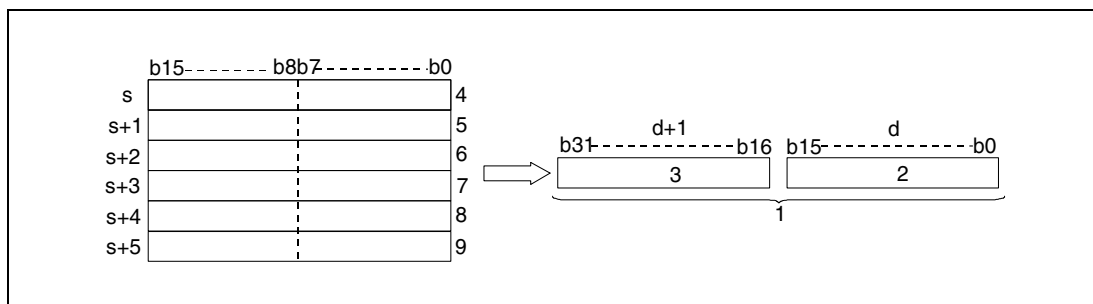
Каждый сохраненный разряд ASCII-кода может принимать значение между "30н" и "39н".

Если разряд содержит значение "20н" или "00н", то этот разряд автоматически записывается в виде кода "30н".



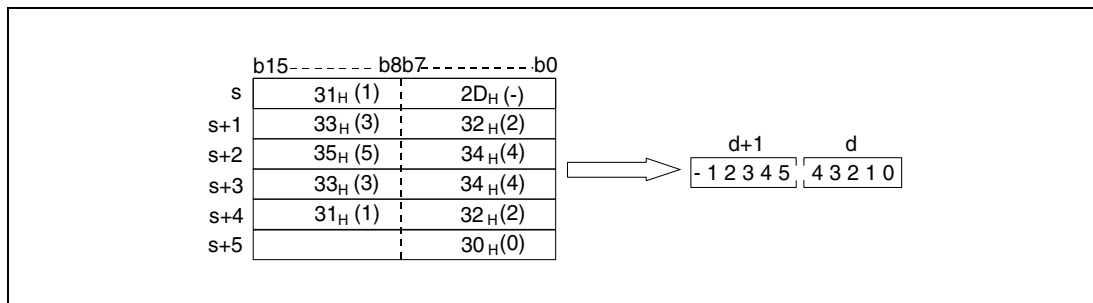
**DDABIN Преобразование в 32-битные двоичные данные**

Команда DDABIN преобразует десятичные данные в кодировке ASCII, указанные в операндах от s (Array\_s[1]) до s + 5 (Array\_s[6]), в 32-битный двоичный формат и сохраняет результат в d и d + 1.



- <sup>1</sup> 32-битные двоичные данные
- <sup>2</sup> младшие 16 битов
- <sup>3</sup> старшие 16 битов
- <sup>4</sup> ASCII-код разряда миллиардов/арифметический знак
- <sup>5</sup> ASCII-код разряда десятков миллионов/ASCII-код разряда сотен миллионов
- <sup>6</sup> ASCII-код разряда сотен тысяч/ASCII-код разряда миллионов
- <sup>7</sup> ASCII-код разряда тысяч/ASCII-код разряда десятков тысяч
- <sup>8</sup> ASCII-код разряда десятков/ASCII-код разряда сотен
- <sup>9</sup> игнорируется/ASCII-код разряда десятков

Значение, указанное в операндах от s (Array\_s[1]) до s + 5 (Array\_s[6]), сохраняется, начиная с d, в виде -1234543210<sub>ВВ</sub> следующим образом:



Значение в кодировке ASCII, указанное в операндах от s (Array\_s[1]) до s + 5 (Array\_s[6]), может находиться в диапазоне между -2147483648 и 2147483647.

Если двоичное значение положительное, арифметический знак сохраняется в виде кода "20<sub>H</sub>".

При отрицательном значении арифметический знак сохраняется в виде кода "2D<sub>H</sub>".

Каждый сохраненный разряд ASCII-кода может принимать значение между "30<sub>H</sub>" и "39<sub>H</sub>".

Если разряд содержит значение "20<sub>H</sub>" или "00<sub>H</sub>", то значение этого разряда автоматически сохраняется в виде кода "30<sub>H</sub>".

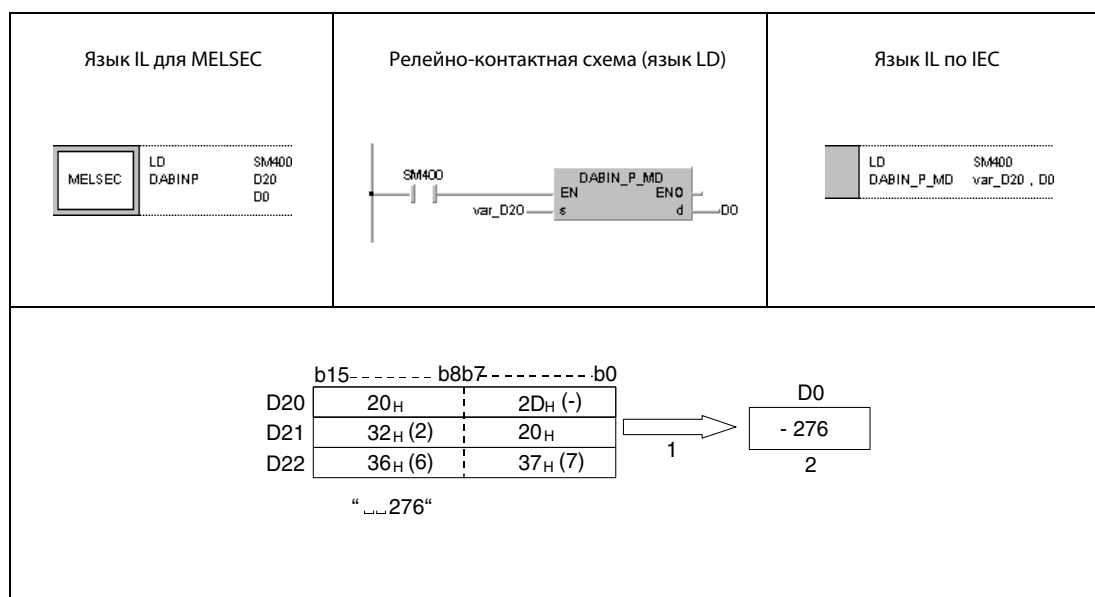
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Арифметический знак в нижних 8 битах операнда s (Array\_s[1]) содержит иное значение кроме "20H" или "2DH" (код ошибки 4100).
- ASCII-код в операндах от s (Array\_s[1]) до s + 5 (Array\_s[6]) содержит иные значения кроме "30H"... "39H", "20H" или "00H" (код ошибки 4100).
- Число в ASCII-кодировке в операндах от s (Array\_s[1]) до s + 5 (Array\_s[6]) находится вне следующих диапазонов:  
 в случае команды DABIN от -32768 до 32767  
 в случае команды DDABIN от -2147483648 до 2147483647 (код ошибки 4100)

**Пример 1** DABINP

Следующая программа при положительном фронте сигнала SM400 преобразует пятизначное десятичное значение ASCII, содержащееся в регистрах от D20 (var\_D20 Array [0]) до D22 (var\_D20 Array [2]), в двоичное значение и сохраняет его в D0.

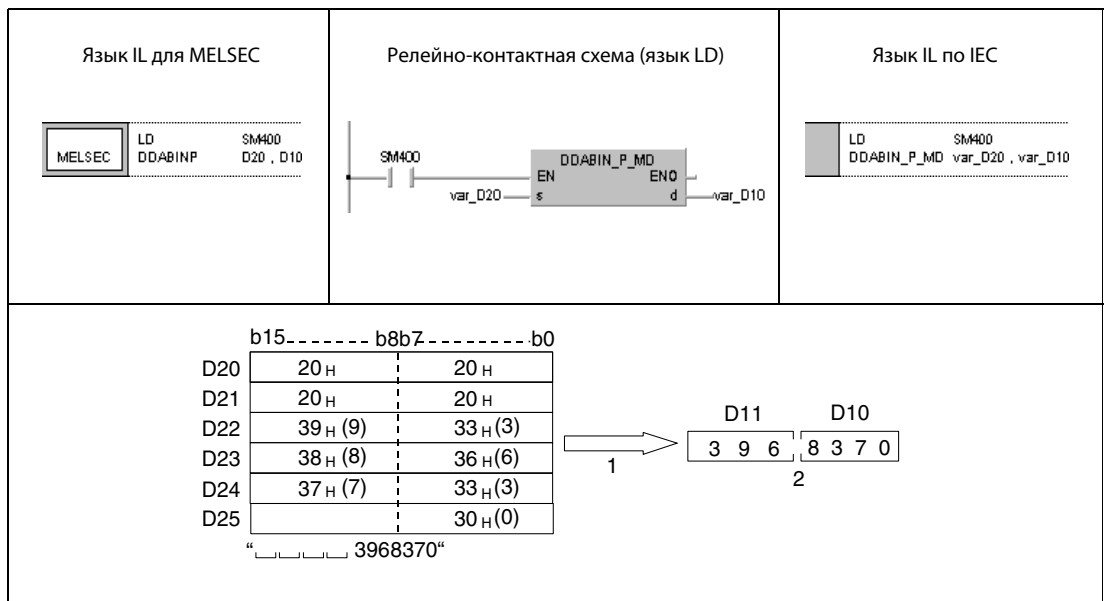


<sup>1</sup> считывается как -00276

<sup>2</sup> двоичное значение

**Пример 2** DDABINP

Следующая программа при положительном фронте сигнала SM400 преобразует десятизначное десятичное значение в формате ASCII, содержащееся в регистрах от D20 (var\_D20 Array [0]) до D25 (var\_D20 Array [5]), в двоичное значение и сохраняет его в D10 и D11.



<sup>1</sup> считывается как +0003968370  
<sup>2</sup> двоичное значение

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

7.11.5 HABIN, HABINP, DHABIN, DHABINP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

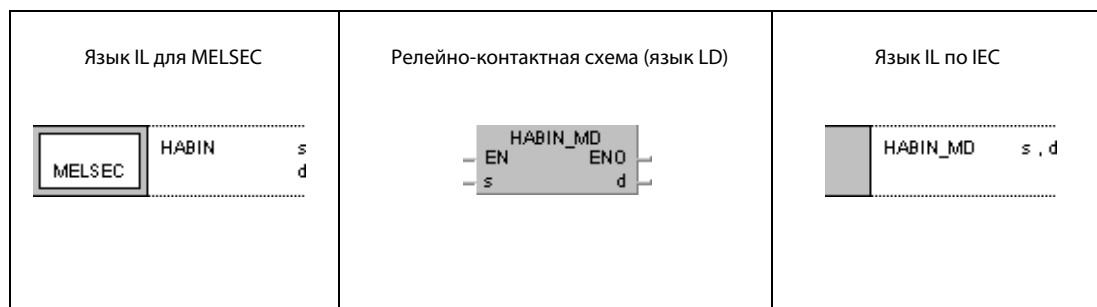
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

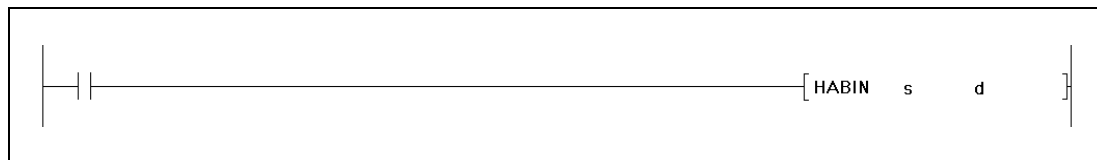
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□NG□□	Индексные регистры Zn	Константы \$			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

GX IEC Developer



GX Developer

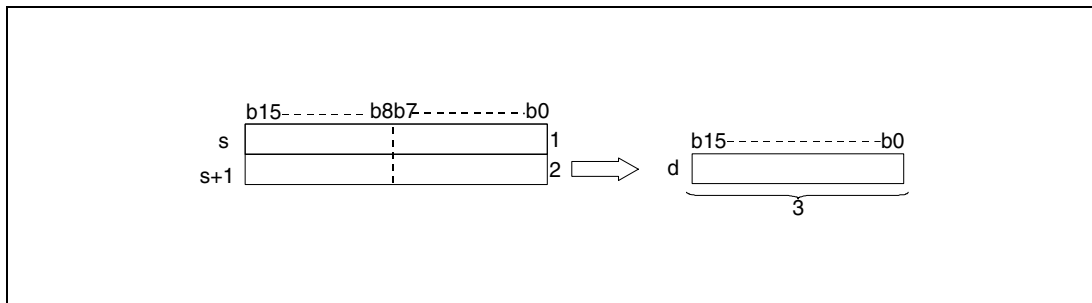


Переменные

Операнд	Значение	Тип данных	
		MELSEC	IEC
s	Область памяти, в которой сохранены конвертируемые данные в формате ASCII.	строковая величина	ANY32/ массив [1..4] данных типа ANY16
d	Область памяти, в которой сохраняется результат конвертации.	BIN, 16/32 бита	ANY16/32

**Принцип действия****Преобразование шестнадцатеричных данных в кодировке ASCII в 16-битные или 32-битные двоичные данные****HABIN Преобразование в 16-битные двоичные данные**

Команда HABIN преобразует шестнадцатеричные данные в кодировке ASCII, указанные в  $s$  и  $s + 1$ , в 16-битные двоичные данные и сохраняет результат в  $d$ .

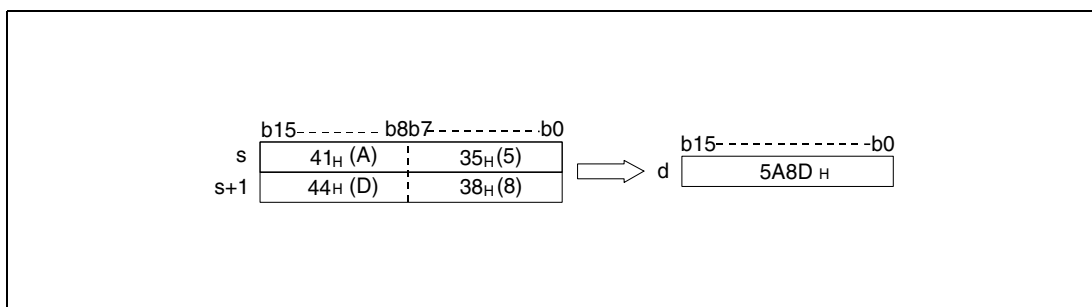


<sup>1</sup> ASCII-код 3-го разряда/ASCII-код 4-го разряда

<sup>2</sup> ASCII-код 1-го разряда/ASCII-код 2-го разряда

<sup>3</sup> 16-битные двоичные данные

После обработки значение "5A8Dн", указанное в данных от  $s$  до  $s + 1$ , сохраняется в  $d$  следующим образом:

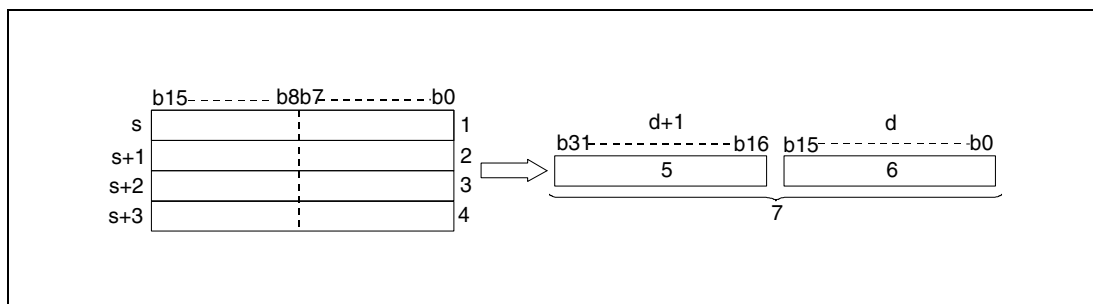


Значение в кодировке ASCII, указанное в данных от  $s$  до  $s + 1$ , может находиться в диапазоне от 0000н до FFFFн.

Каждый сохраненный разряд ASCII-кода может принимать значение между "30н" и "39н", "41н" и "46н".

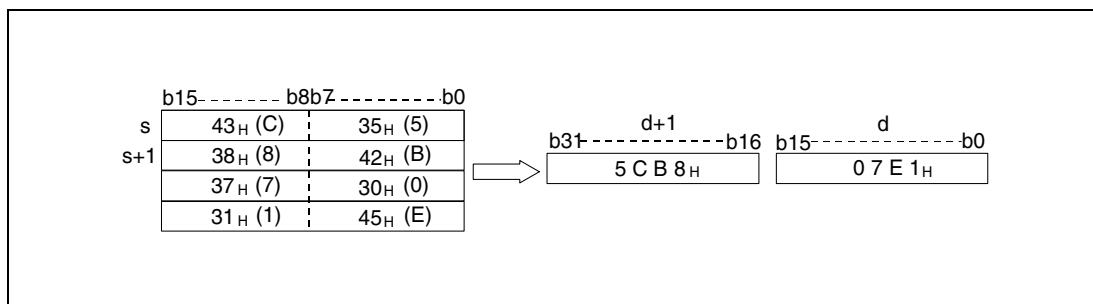
**DHABIN Преобразование в 32-битные двоичные данные**

Команда DHABIN преобразует шестнадцатеричные данные в кодировке ASCII, указанные в области от s (Array\_s[1]) до s+3 (Array\_s[4]), в 32-битный двоичный формат и сохраняет результат в d и d+1.



- <sup>1</sup> ASCII-код 7-го разряда/ASCII-код 8-го разряда
- <sup>2</sup> ASCII-код 5-го разряда/ASCII-код 6-го разряда
- <sup>3</sup> ASCII-код 3-го разряда/ASCII-код 4-го разряда
- <sup>4</sup> ASCII-код 1-го разряда/ASCII-код 2-го разряда
- <sup>5</sup> старшие 16 битов
- <sup>6</sup> младшие 16 битов
- <sup>7</sup> 32-битные двоичные данные

Значение "5CB807E1", указанное в области от s (Array\_s[1]) до s + 3 (Array\_s[4]), после обработки сохраняется в d и d + 1 следующим образом:



Значение в кодировке ASCII, указанное в области от s (Array\_s[1]) до s + 3 (Array\_s[4]), может находиться в диапазоне от 00000000н до FFFFFFFFн.

Каждый сохраненный разряд ASCII-кода может принимать значение между "30н" и "39н", "41н" и "46н".

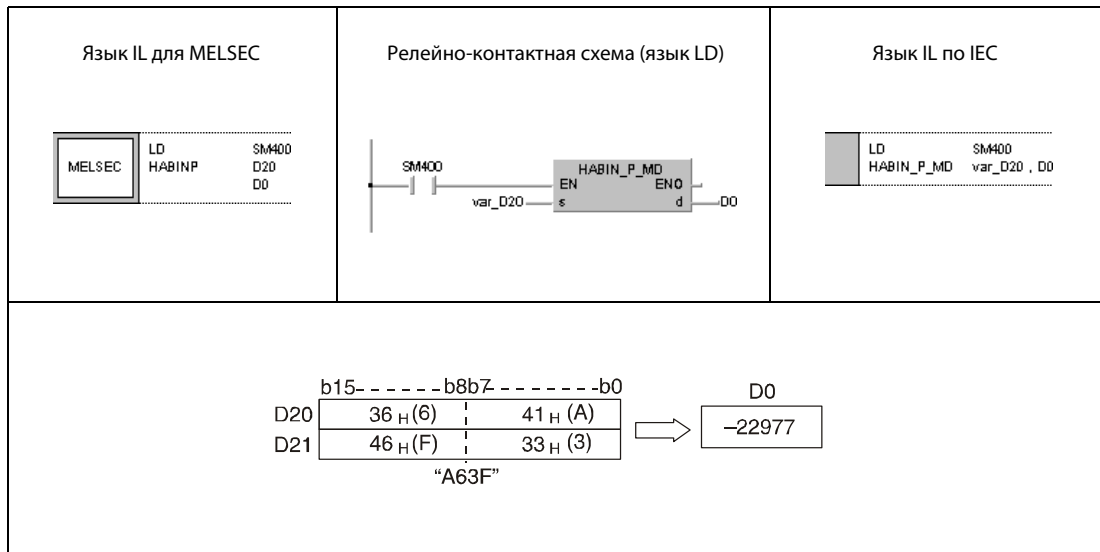
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- ASCII-код в области от s (Array\_s[1]) до s + 3 (Array\_s[4]) находится вне диапазона от "30н" до "39н" и от "41н" до "46н" (код ошибки 4100).

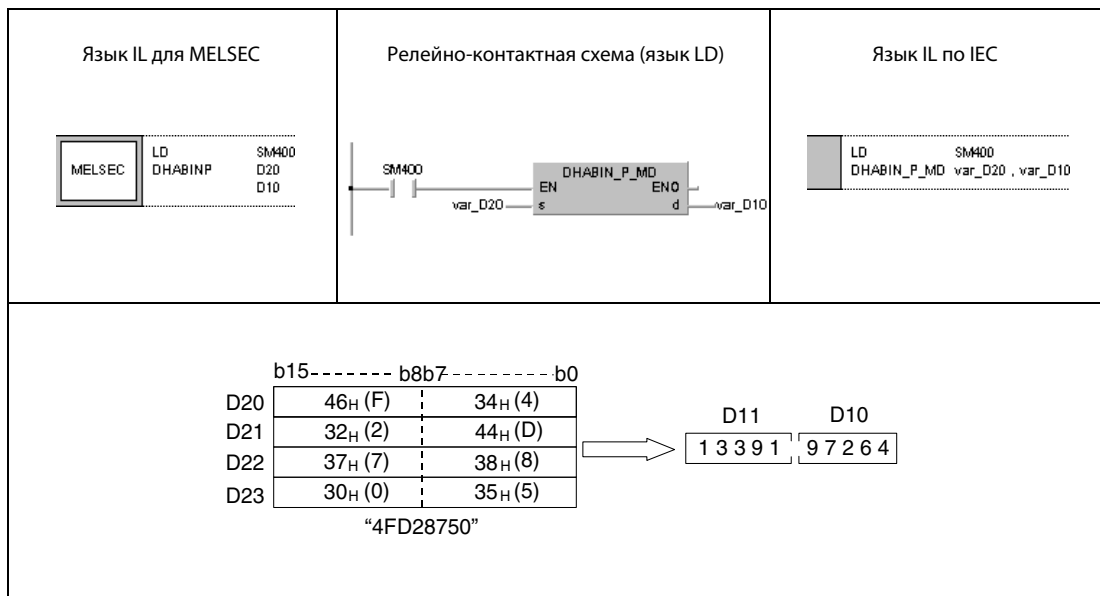
**Пример 1** HABINP

Следующая программа при положительном фронте сигнала SM400 преобразует 4-значное значение в кодировке ASCII, хранящееся в области от D20 (var\_D20 Array [0]) до D21 (var\_D20 Array [1]), в двоичное значение и сохраняет его в D0.



**Пример 2** DHABINP

Следующая программа при положительном фронте сигнала SM400 преобразует 8-значное значение в кодировке ASCII, хранящееся в области от D20 (var\_D20 Array [0]) до D23 (var\_D20 Array [3]), в двоичное значение и сохраняет его в D10 и D11.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.11.6 DABCD, DABCDP, DDABCD, DDABCDP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

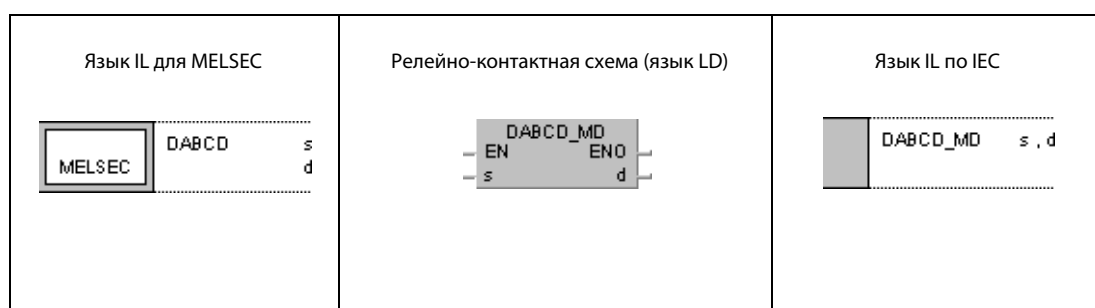
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

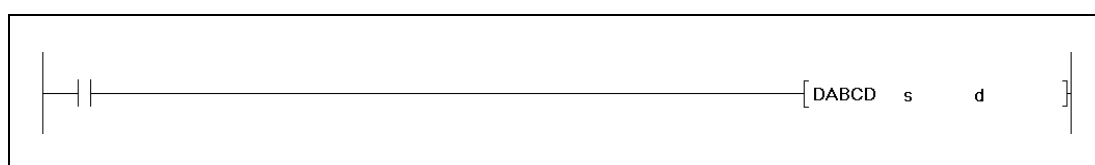
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы \$			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных	
		MELSEC	IEC
s	Область памяти, в которой сохранены конвертируемые данные в формате ASCII.	строковая величина	ANY32/ массив [1..4] данных типа ANY16
d	Область памяти, в которой сохраняется результат конвертации.	4-значные и 8-значные двоично-десятичные данные	ANY16/32

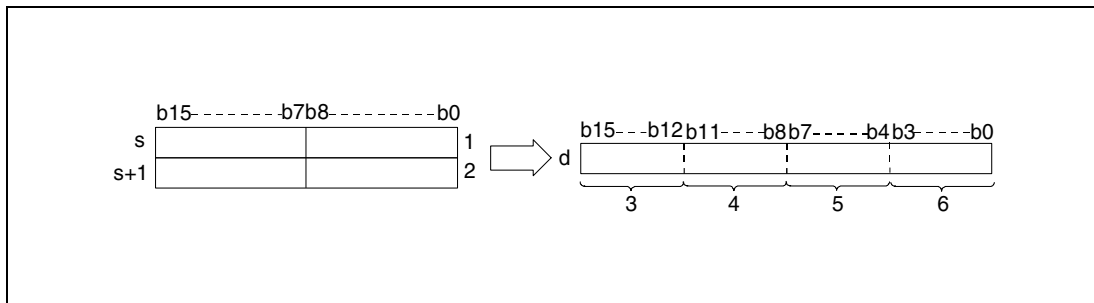


**Принцип действия**

**Преобразование десятичных данных в кодировке ASCII в 4-значные и 8-значные двоично-десятичные данные**

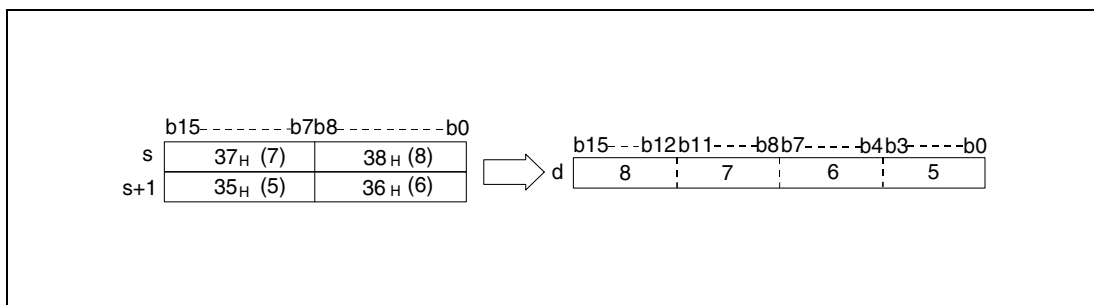
**DABCD Преобразование в 4-значные двоично-десятичные данные**

Команда DABCD преобразует десятичные данные в кодировке ASCII, указанные в s и s + 1, в 4-значные двоично-десятичные данные и сохраняет их в d.



- <sup>1</sup> ASCII-код разряда сотен/ASCII-код разряда тысяч
- <sup>2</sup> ASCII-код разряда единиц/ASCII-код разряда десятков
- <sup>3</sup> разряд тысяч
- <sup>4</sup> разряд сотен
- <sup>5</sup> разряд десятков
- <sup>6</sup> разряд единиц

Указанное в s и s + 1 значение 8765 сохраняется в d следующим образом:



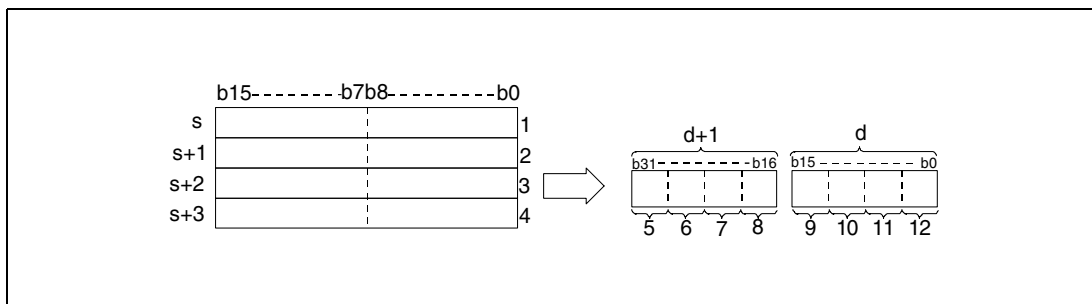
Значение в кодировке ASCII, указанное в области от s до s + 1, может находиться в диапазоне от 0 до 9999.

Каждый сохраненный разряд ASCII-кода может принимать значение между "30н" и "39н".

Если разряд содержит значение "20н" или "00н", оно автоматически перезаписывается значением "30н".

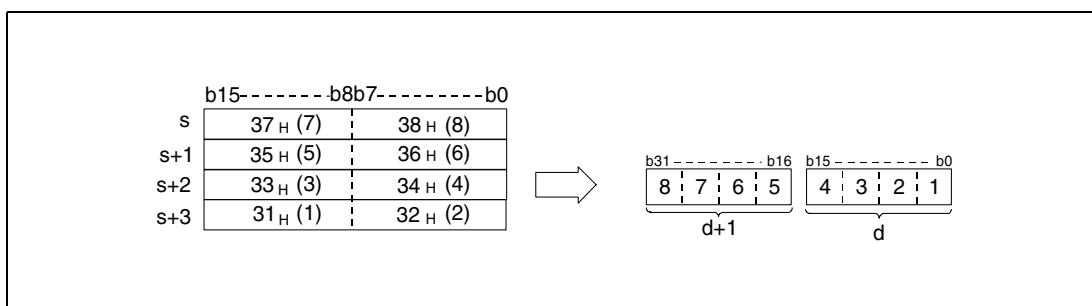
**DDABCD Преобразование в 8-значные двоично-десятичные данные**

Команда DDABCD преобразует десятичные данные в кодировке ASCII, указанные в области от s (Array\_s[1]) до s + 3 (Array\_s[4]), в 8-значный двоично-десятичный формат и сохраняет результат в d и d + 1.



- <sup>1</sup> ASCII-код разряда миллионов/ASCII-код разряда десятков миллионов
- <sup>2</sup> ASCII-код разряда десятков тысяч/ASCII-код разряда сотен тысяч
- <sup>3</sup> ASCII-код разряда сотен/ASCII-код разряда тысяч
- <sup>4</sup> ASCII-код разряда единиц/ASCII-код разряда десятков
- <sup>5</sup> разряд десятков миллионов
- <sup>6</sup> разряд миллионов
- <sup>7</sup> разряд сотен тысяч
- <sup>8</sup> разряд десятков тысяч
- <sup>9</sup> разряд тысяч
- <sup>10</sup> разряд сотен
- <sup>11</sup> разряд десятков
- <sup>12</sup> разряд единиц

Значение 87654321, указанное в области от s (Array\_s[1]) до s + 3 (Array\_s[4]), сохраняется в d и d + 1 следующим образом:



Значение в кодировке ASCII, указанное в области от s (Array\_s[1]) до s + 3 (Array\_s[4]), может находиться в диапазоне от 0 до 99999999.

Каждый сохраненный разряд ASCII-кода может принимать значение между "30н" и "39н".

Если разряд содержит значение "20н" или "00н", оно автоматически перезаписывается значением "30н".

**Источники ошибок**

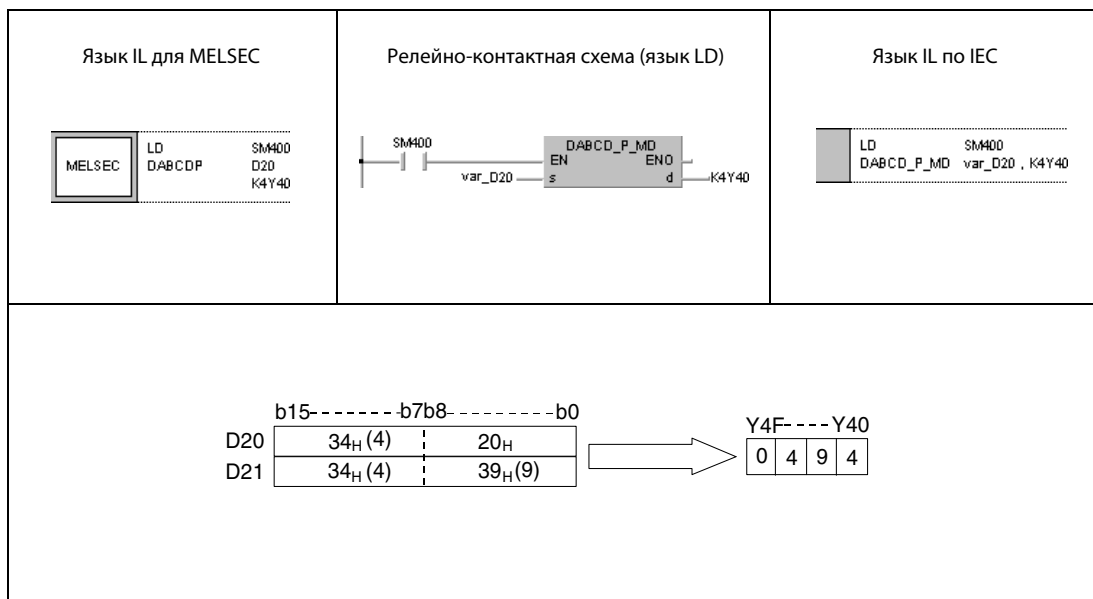
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- ASCII-код в области от s (Array\_s[1]) до s + 3 (Array\_s[4]) находится вне допустимого диапазона от "30н" до "39н" (код ошибки 4100).

**Пример 1**

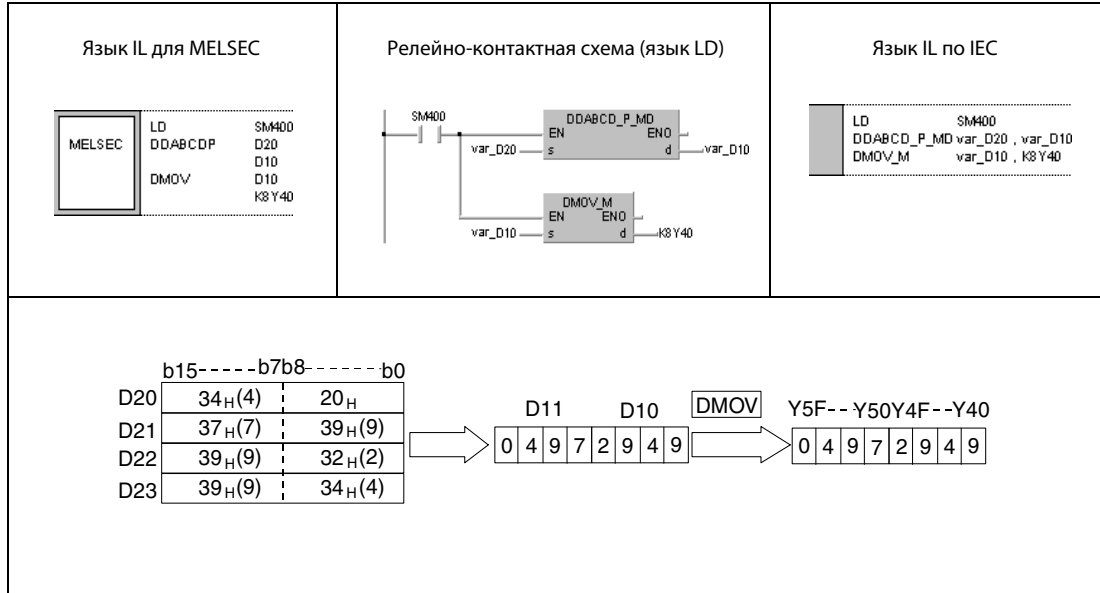
**DABCDP**

Следующая программа при положительном фронте сигнала SM400 преобразует значение в кодировке ASCII, хранящееся в области от D20 (var\_D20 Array [0]) до D21 (var\_D20 Array [1]), в 4-значное двоично-десятичное значение и выводит его на выходы от Y40 до Y4F.



**Пример 2** DDABCDP

Следующая программа при положительном фронте сигнала SM400 преобразует значение в кодировке ASCII, хранящееся в области от D20 (var\_D20 [0]) до D23 (var\_D20 [3]), в 8-значное двоично-десятичное значение, сохраняет результат по адресу D10 и D11 и выводит его на выходы от Y40 до Y5F.



### 7.11.7 COMRD, COMRDP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

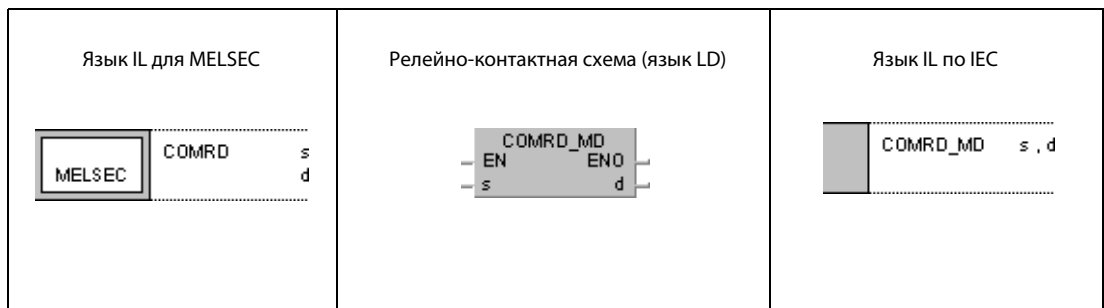
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

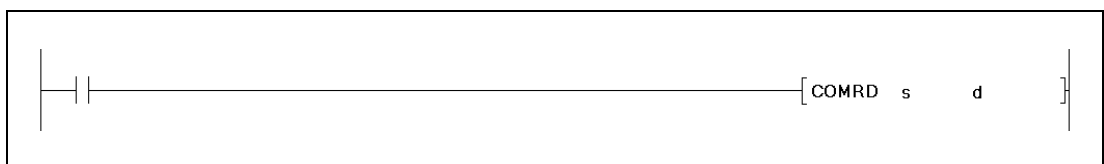
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)	Иные BL\S, BL\TR, BL, P, I, J, U		
	битовые	словные		битовые	словные						
s	●	●	●	●	●	—	—	—	●	SMO	3
d	—	●	●	—	—	—	—	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

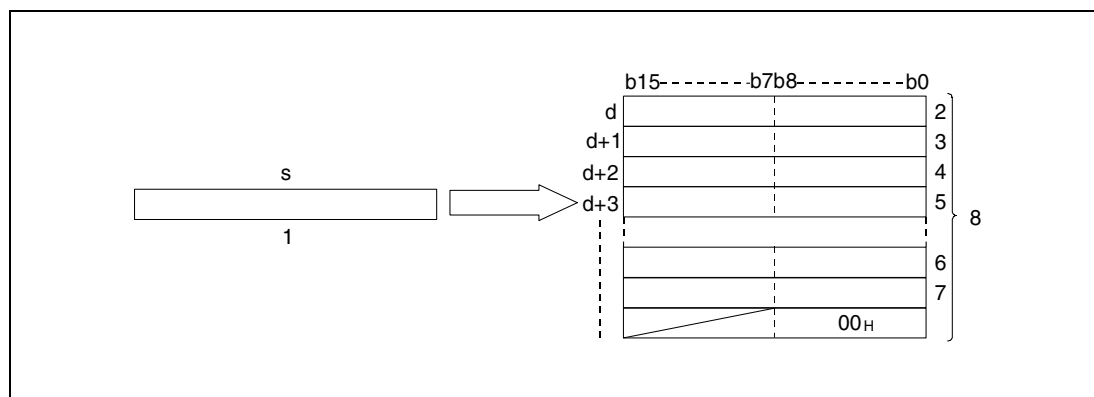
Операнд	Значение	Тип данных	
		MELSEC	IEC
s	Первый адрес операнда, из которого требуется считать сохраненный комментарий.	адрес	ANY16
d	Первый адрес операнда, в котором требуется сохранить данные комментариев.	строковая величина	массив [1..8] данных типа ANY16

**Принцип действия**

**Считывание данных комментариев**

**COMRD Команда считывания**

Команда COMRD считывает данные комментариев из *s* и сохраняет их в ASCII-кодировке в области от *d* (Array\_d[1]) до *d* + 7 (Array\_d[8]).



<sup>1</sup> комментарий

<sup>2</sup> ASCII-код 2-го знака/ASCII-код 1-го знака

<sup>3</sup> ASCII-код 4-го знака/ASCII-код 3-го знака

<sup>4</sup> ASCII-код 6-го знака/ASCII-код 5-го знака

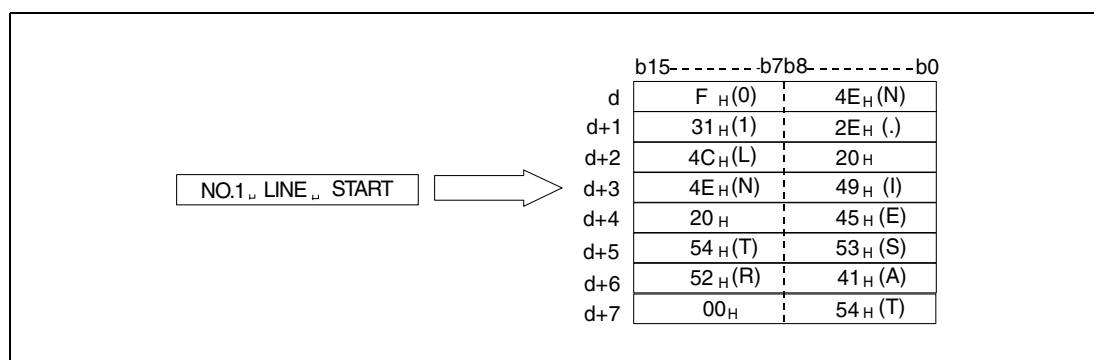
<sup>5</sup> ASCII-код 8-го знака/ASCII-код 7-го знака

<sup>6</sup> ASCII-код 30-го знака/ASCII-код 29-го знака

<sup>7</sup> ASCII-код 32-го знака/ASCII-код 31-го знака

<sup>8</sup> сохраняет максимум 32 знака

Хранящийся в *s* комментарий со строковой величиной "NO.1 LINE START" сохраняется, начиная с *d*, следующим образом (Array\_d[1]):



Область адресов в *s* должна находиться в пределах диапазона адресов для комментариев.

Если в *s* не указан комментарий, знаки преобразуются в знаки пробела.

Размер комментария не должен превышать 32 знака.

Содержимое байта после последнего знака зависит от состояния маркера диагностики SM701:

Если SM701 не установлен, сохраняется ноль.

Если SM701 установлен, никакие изменения не предпринимаются.

В процессорах серии "Q" или "System Q" после выполнения этой команды на один цикл устанавливается маркер SM720. Во время выполнения команды устанавливается маркер SM721. Если SM721 установлен, обработка команды COMRD(P) не возможна.

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Область адресов в *s* находится вне области комментария (код ошибки 4100).
- Команда COMRD(P) выполняется в тот момент, когда в режиме RUN записывается комментарий (код ошибки 4100).
- Выбранный файл не существует (код ошибки 2410).

**ПРИМЕЧАНИЯ**

*Процессор "System Q" выполняет эту команду за несколько прогонов программы. Процессор QnA выполняет команду сразу.*

*Перед выполнением команды COMRD(P)/PRC должна быть завершена прежняя команда COMRD(P)/PRC (маркер SM720 должен иметь состояние "1").*

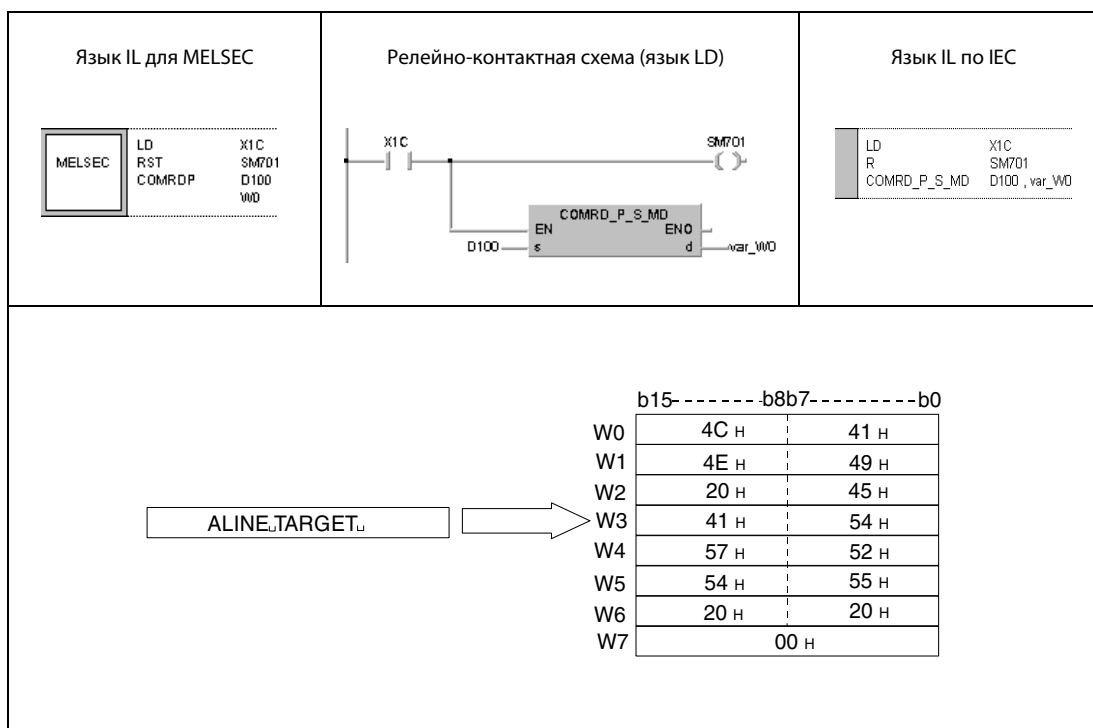
*Одновременное обращение к двум или более комментариям не возможно.*

*Следующие команды не могут выполняться одновременно, так как они используют специальный маркер SM721:*

Команда	Установлен во время выполнения	Установлен на один цикл после выполнения	Установлен после выполнения с ошибкой
S.FREAD S.FWRITE	SM721	бит, указанный в команде	бит, указанный в команде, + следующий бит
PRC COMRD		SM720	—

**Пример****COMRDP**

Следующая программа при положительном фронте сигнала X1C сохраняет комментарий, хранящийся в D100 в кодировке ASCII, в операндах от W0 (var\_W0 Array [0]) до W7 (var\_W0 Array [7]).

**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.



### 7.11.8 LEN, LENP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

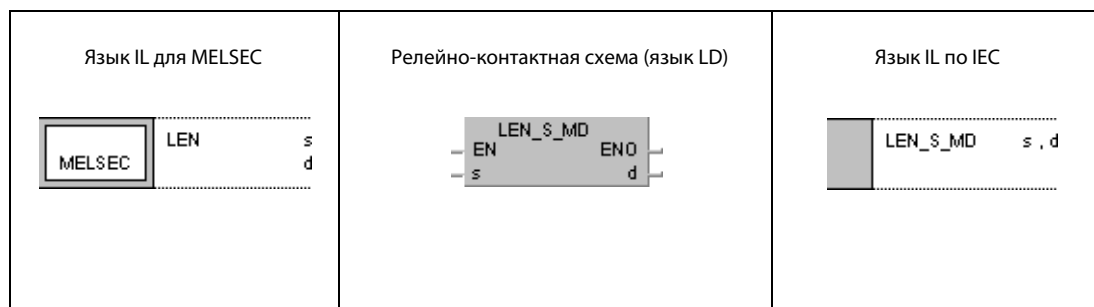
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

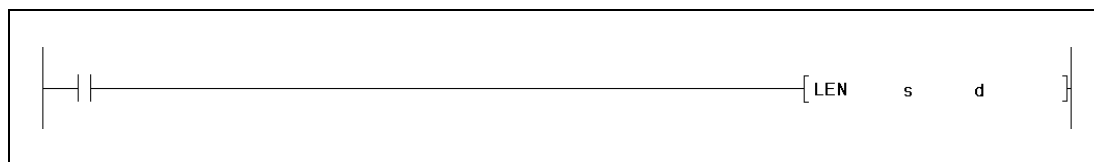
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы \$			Иные U
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

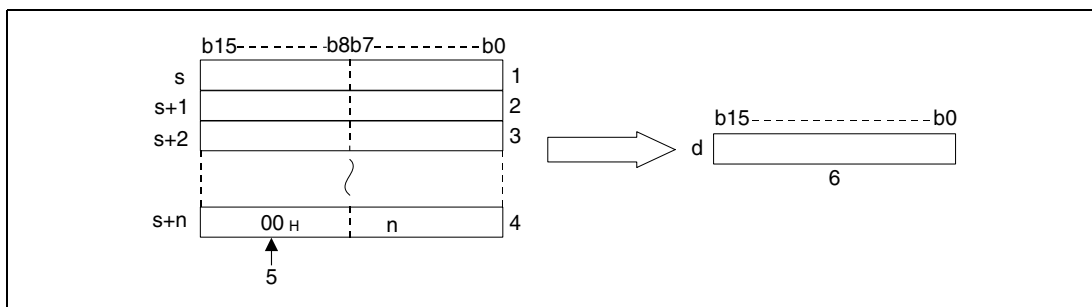
Операнд	Значение	Тип данных
s	Первый адрес операнда, содержащего строковую величину, длину которой требуется определить.	строковая величина
d	Область адресов, в которой сохраняется измеренная длина строковой величины.	BIN, 16 бит

**Принцип действия**

**Определение длины строковых величин**

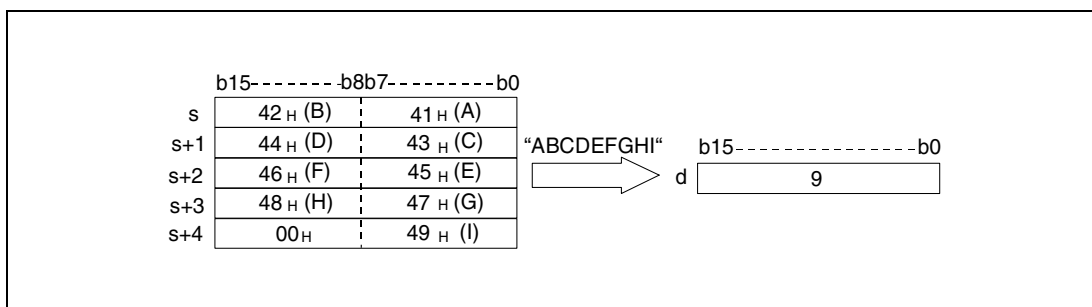
**LEN Измерение длины**

Команда LEN определяет длину строковой величины, указанной в s, и сохраняет результат в d.



- <sup>1</sup> 2-й знак/1-й знак
- <sup>2</sup> 4-й знак/3-й знак
- <sup>3</sup> 6-й знак/5-й знак
- <sup>4</sup> n-ный знак
- <sup>5</sup> конец строковой величины
- <sup>6</sup> длина строковой величины

Результат для строковой величины "ABCDEFGH", хранящейся в s, сохраняется в d в виде числа 9 следующим образом:



Строковая величина, хранящаяся в s, обрабатывается до достижения кода знака "00H".

Результат сохраняется в d.

**Источники ошибок**

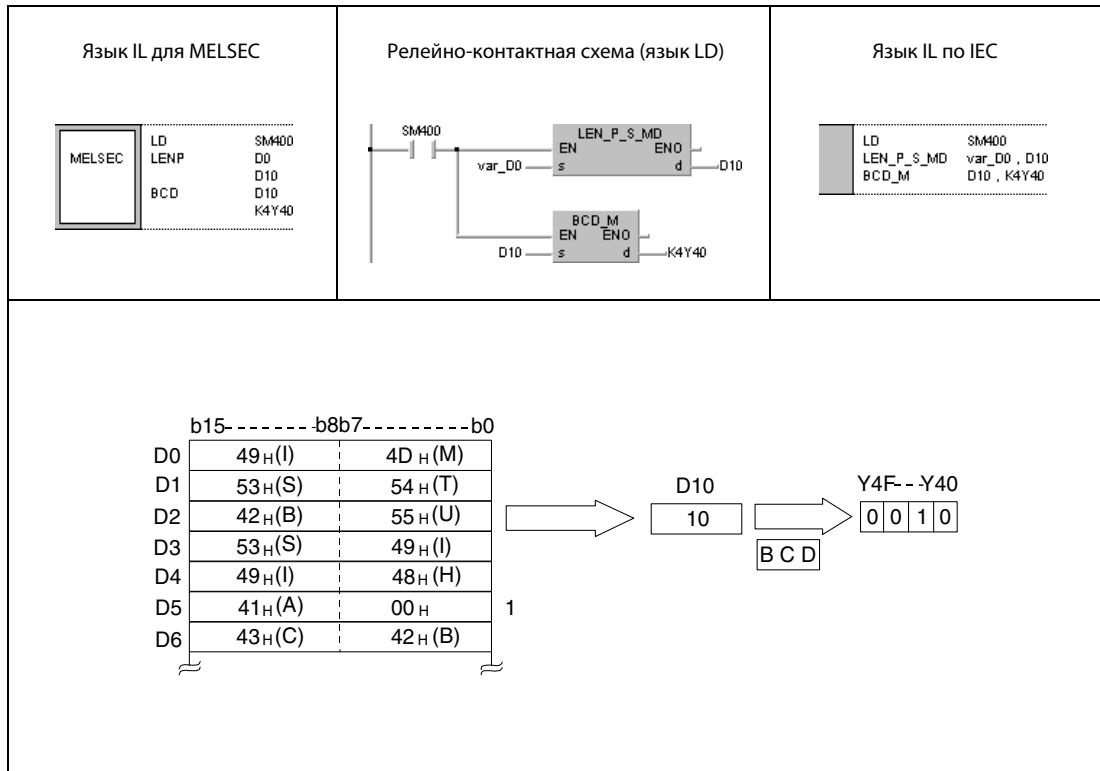
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В последнем байте начиная с s отсутствует код знака "00H" (код ошибки 4101).

**Пример**

LENP

Следующая программа при положительном фронте сигнала SM400 определяет длину строковой величины, хранящейся, начиная с D0 до кода "00н". Результат выводится на выходы Y40–Y4F.



<sup>1</sup> Знаки после кода "00н" игнорируются (определяется только длина строковой величины "MITSUBISHI").

**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

7.11.9 STR, STRP, DSTR, DSTRP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

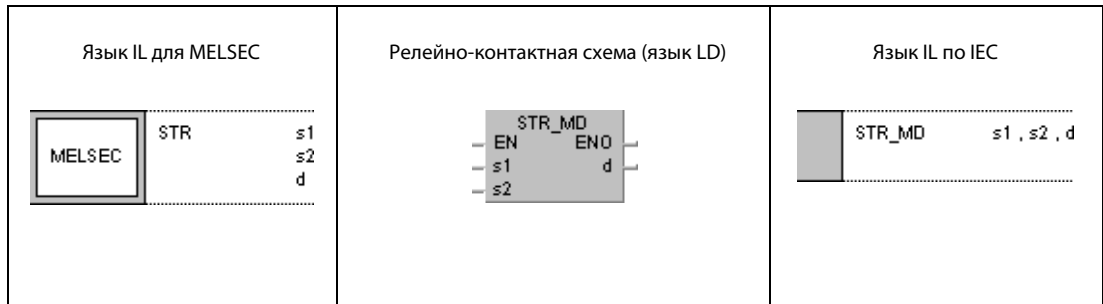
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□NG□□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	●	●	●	●	●	●	●	—	—	SM0	4
s2	●	●	●	●	●	●	●	—	—		
d	—	●	●	—	—	—	—	—	—		

GX IEC Developer



GX Developer



Переменные

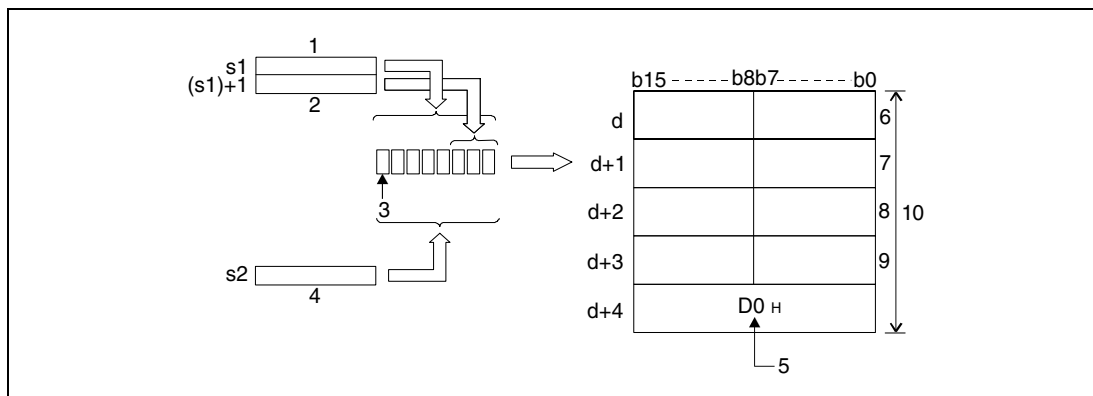
Операнд	Значение	Тип данных	
		MELSEC	IEC
s1	Первый адрес операнда, в котором сохранено количество разрядов конвертируемого числового значения.	BIN, 16 бит	ANY32
s2	Конвертируемые двоичные данные.	BIN, 16/32 бита	ANY16/32
d	Первый адрес операнда, в котором сохраняется конвертированная строковая величина.	строковая величина	массив [1..5]/[1..6] данных типа ANY16

**Принцип действия**

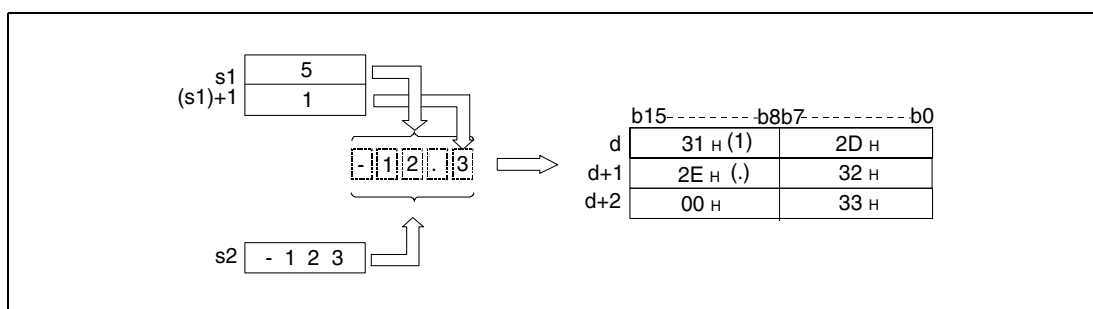
**Конвертирование 16/32-битных двоичных данных в строковые величины**

**STR Преобразование 16-битных двоичных данных**

Команда STR вставляет десятичную запятую в 16-битное двоичное значение, указанное в s2, в месте, указанном в s1 и (s1) + 1. Результат преобразуется в строковую величину и сохраняется в области от d (Array\_d[1]) до d + 4 (Array\_d[5]).



- 1 общее количество разрядов
- 2 дробная часть
- 3 арифметический знак
- 4 двоичное значение
- 5 автоматически установленный конец строковой величины
- 6 ASCII-код знака (общее количество разрядов – 1)/ASCII-код арифметического знака
- 7 ASCII-код знака (общее количество разрядов – 3)/ASCII-код знака (общее количество разрядов – 2)
- 8 ASCII-код знака (общее количество разрядов – 5)/ASCII-код знака (общее количество разрядов – 4)
- 9 ASCII-код знака (общее количество разрядов – 7)/ASCII-код знака (общее количество разрядов – 6)
- 10 общее количество разрядов



Количество разрядов, указанное в s1, должно быть в диапазоне между 2 и 8.

Количество разрядов дробной части, указанное в (s1) + 1, должно быть в диапазоне между 0 и 5 и не должно превышать общее количество разрядов минус 3.

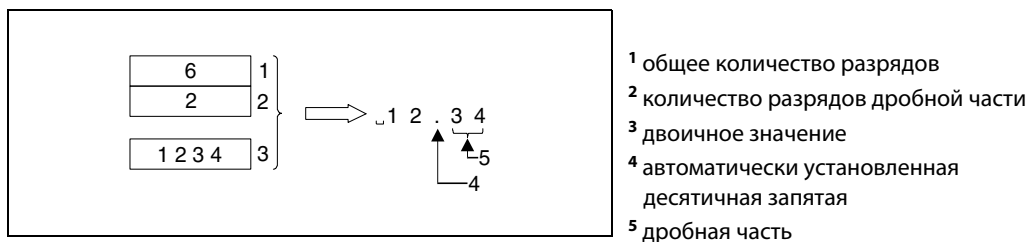
16-битные двоичные данные в s2 должны находиться в диапазоне между –32768 и 32767.

После конвертации строковая величина сохраняется в области от d (Array\_d[1]) до d + 4 (Array\_d[5]) следующим образом:

В качестве положительного арифметического знака двоичных данных используется знак ASCII "20H" (знак пробела).

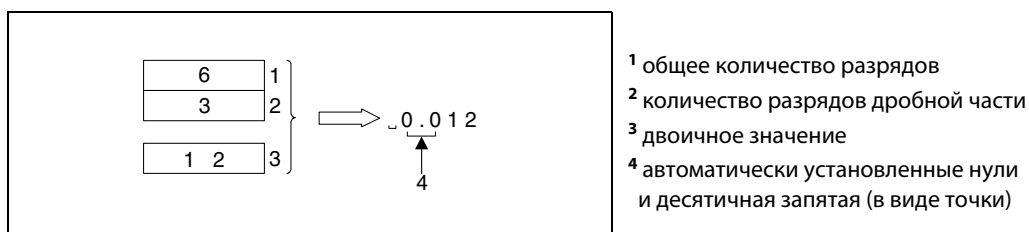
В качестве отрицательного арифметического знака двоичных данных используется знак ASCII "2DH" (знак минус).

Если количество разрядов дробной части больше нуля, десятичная запятая "2Ен" (в виде точки ".") автоматически устанавливается перед первым указанным разрядом.

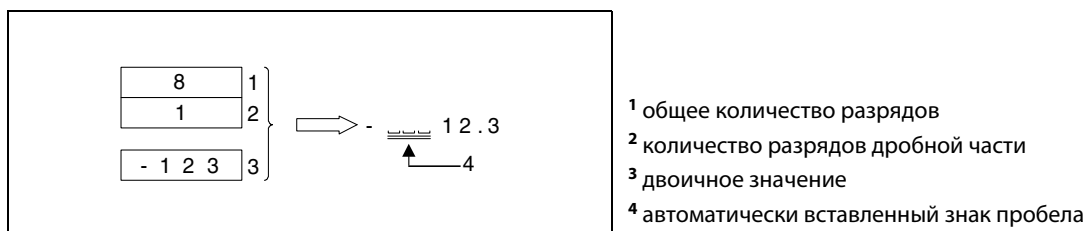


Если количество разрядов дробной части равна нулю, знак десятичной запятой "2Дн" не устанавливается.

Если количество разрядов дробной части больше количества разрядов двоичного значения, то недостающие разряды автоматически заполняются нулями, число двоичного значения смещается вправо и в соответствующем месте вставляется десятичная запятая (0.□□□□□).



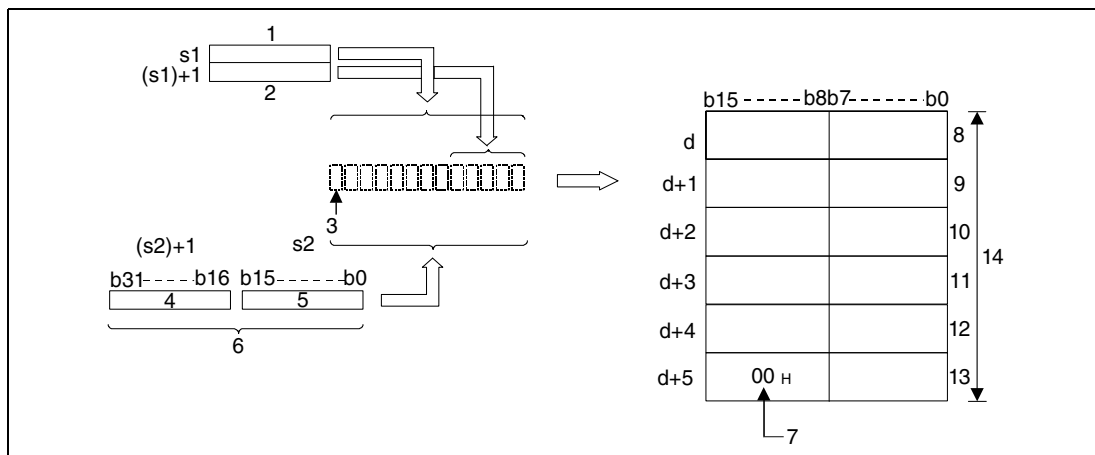
Если общее количество разрядов, включая арифметический знак и десятичную запятую, больше количества разрядов двоичного значения, то недостающие разряды между арифметическим знаком и числовым значением автоматически заполняются знаками пробела "20н".



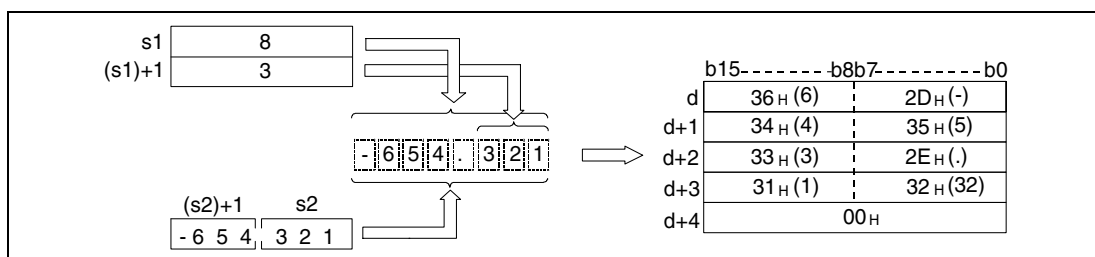
В конце конвертированной строковой величины автоматически сохраняется код "00н".

**DSTR Преобразование 32-битных двоичных данных**

Команда DSTR вставляет в 32-битное двоичное значение, хранящееся в s2 и (s2) + 1, десятичную запятую в месте, указанном в s1 и (s1) + 1. Результат преобразуется в строковую величину и сохраняется в области от d (Array\_d[1]) до d + 5 (Array\_d[6]).



- 1 общее количество разрядов
- 2 дробная часть
- 3 арифметический знак
- 4 старшие 16 битов
- 5 младшие 16 битов
- 6 двоичное значение
- 7 автоматически установленный конец строковой величины
- 8 ASCII-код знака (общее количество разрядов – 1)/ASCII-код арифметического знака
- 9 ASCII-код знака (общее количество разрядов – 3)/ASCII-код знака (общее количество разрядов – 2)
- 10 ASCII-код знака (общее количество разрядов – 5)/ASCII-код знака (общее количество разрядов – 4)
- 11 ASCII-код знака (общее количество разрядов – 7)/ASCII-код знака (общее количество разрядов – 6)
- 12 ASCII-код знака (общее количество разрядов – 9)/ASCII-код знака (общее количество разрядов – 8)
- 13 обозначение конца строковой величины/ASCII-код знака (общее количество разрядов – 10)
- 14 общее количество разрядов



Общее количество разрядов, указанное в s1, должно быть в диапазоне между 2 и 13.

Количество разрядов дробной части, указанное в (s1) + 1, должно быть в диапазоне между 0 и 10 и не должно превышать общее количество разрядов минус 3.

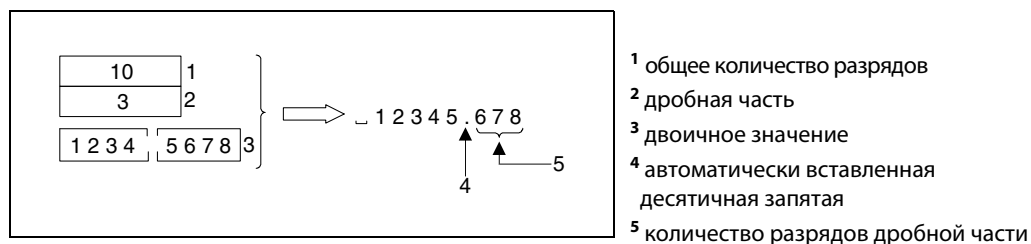
32-битные двоичные данные, сохраняемые в s2 и (s2) + 1, должны быть в диапазоне между -2147483648 и 32147483647.

После преобразования строковая величина сохраняется в области от d (Array\_d[1]) до d + 5 (Array\_d[6]) следующим образом:

В качестве положительного арифметического знака двоичных данных используется знак ASCII "20h" (знак пробела).

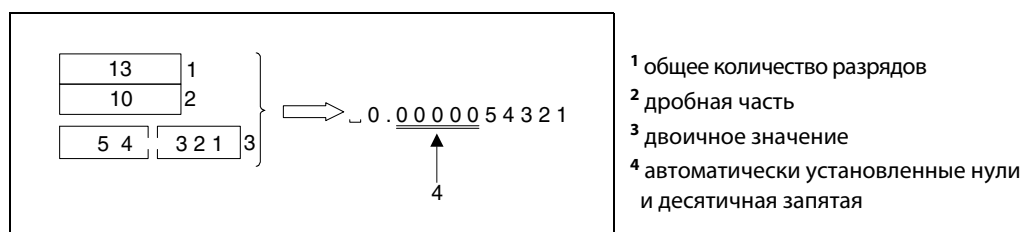
В качестве отрицательного арифметического знака двоичных данных используется знак ASCII "2Dh" (знак минус).

Если количество разрядов дробной части больше нуля, то перед первым указанным разрядом автоматически вставляется десятичная запятая "2Eh" (в виде точки ".").

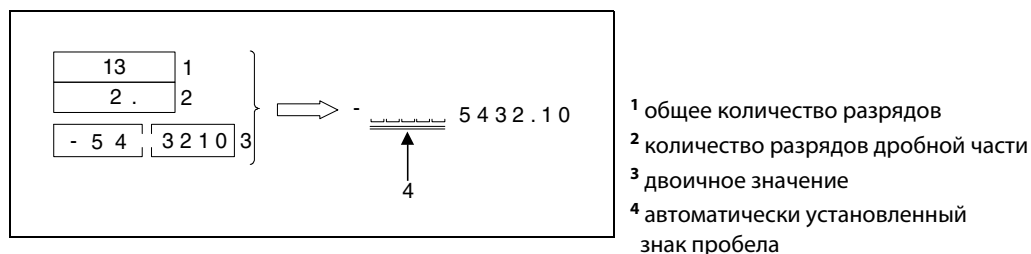


Если количество разрядов дробной части равно нулю, десятичная запятая "2Dh" не вставляется.

Если количество разрядов дробной части больше количества разрядов двоичного значения, то недостающие разряды автоматически заполняются нулями, число двоичного значения смещается вправо и в соответствующем месте вставляется десятичная запятая (0.□□□□).



Если общее количество разрядов, включая арифметический знак и десятичную запятую, больше количества разрядов двоичного значения, то недостающие разряды между арифметическим знаком и числовым значением автоматически заполняются знаками пробела "20h".



В конце конвертированной строковой величины автоматически сохраняется код "00h".



**Источники ошибок**

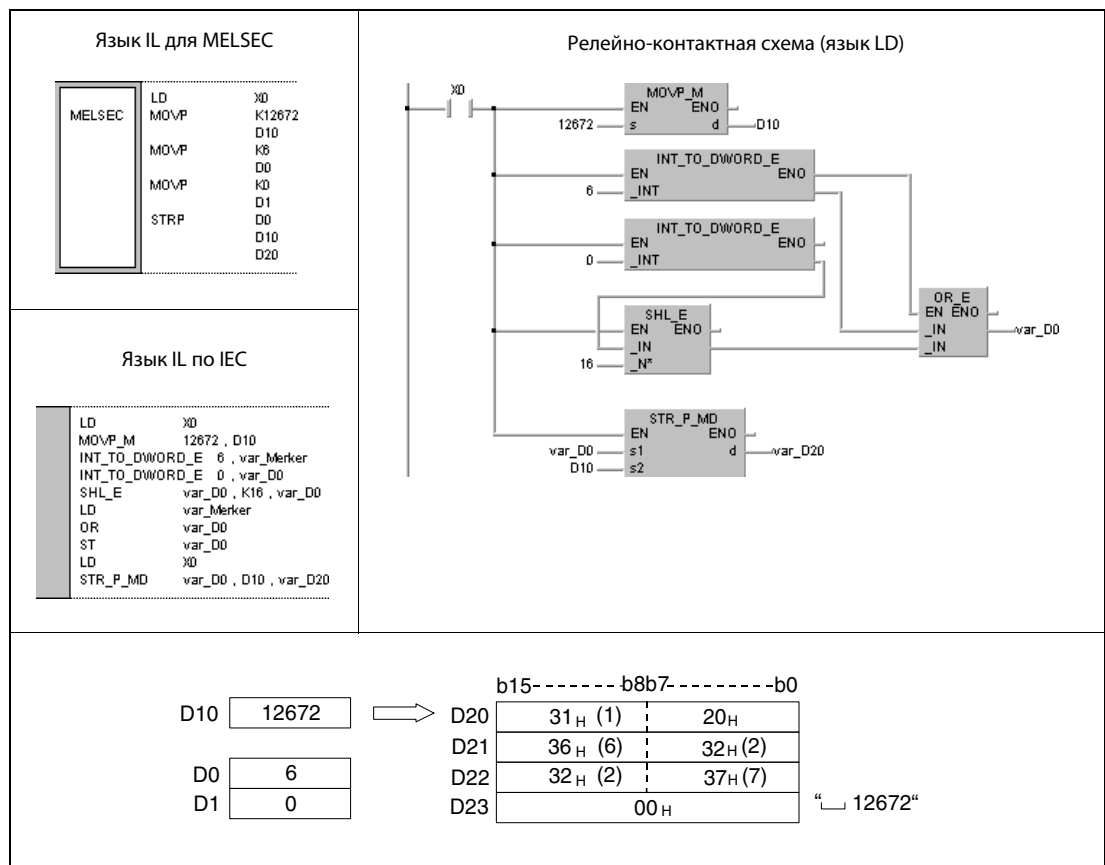
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в s1 общее количество разрядов находится вне нижеуказанных диапазонов (код ошибки 4100).  
диапазон для команды STR.....от 2 до 8  
диапазон для команды DSTR.....от 2 до 13
- Указанное в (s1) + 1 количество разрядов дробной части находится вне нижеуказанных диапазонов (код ошибки 4100).  
диапазон для команды STR.....от 0 до 5  
диапазон для команды DSTR.....от 0 до 10
- Указанные в s1 и (s1) + 1 значения не отвечают следующему соотношению:  
Общее количество разрядов минус 3 превышает или равно количеству разрядов дробной части (код ошибки 4100).
- Количества разрядов, указанные в s1 и (s1) + 1, меньше количества разрядов двоичного значения в s2 и (s2) + 1 (код ошибки 4100).
- Область памяти, начиная с d (Array\_d[1]), в которой сохраняется строковая величина, находится вне предусмотренного для сохранения диапазона (код ошибки 4100).

**Пример 1**

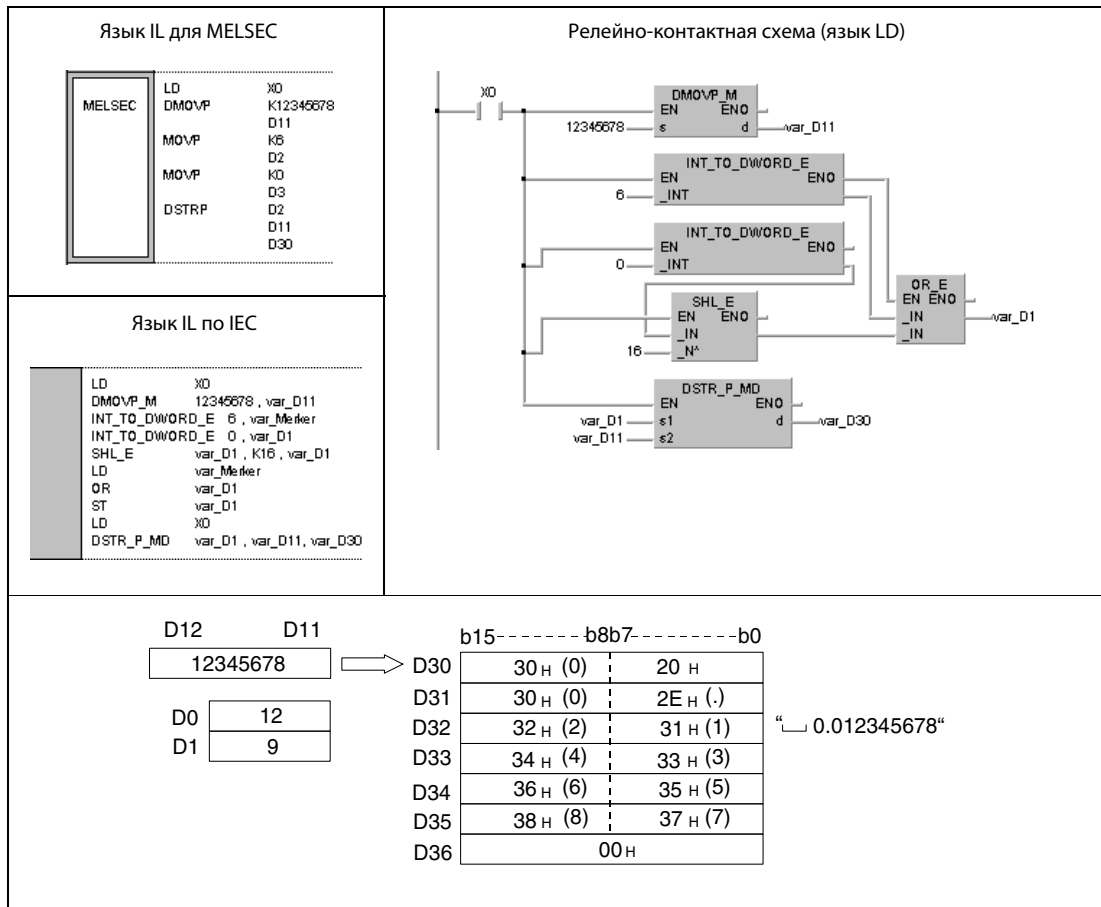
STRP

Следующая программа при положительном фронте сигнала X0 преобразует указанное в D10 двоичное значение в соответствии с количествами разрядов, указанными в D0 и D1, и сохраняет результат в области от D20 (var\_D20 Array [1]) до D23 (var\_D20 Array [4]).



**Пример 2** DSTRP

Следующая программа при положительном фронте сигнала X0 преобразует указанное в D11 и D12 двоичное значение в соответствии с количествами разрядов, указанными в D0 и D1, и сохраняет результат в области от D30 (var\_D20 Array [1]) до D36 (var\_D20 Array [7]).



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.11.10 VAL, VALP, DVAL, DVALP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

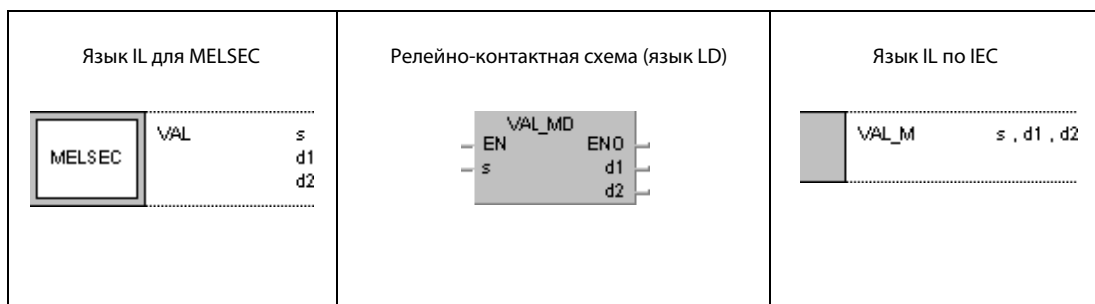
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

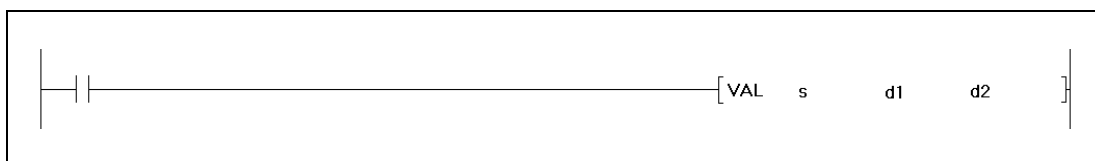
**Операнды MELSEC Q**

Операнды									Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы \$	Иные		
битовые	словные		битовые	словные						
—	●	●	—	—	—	—	●	—	SM0	4
●	●	●	—	—	—	—	—	—		
●	●	●	●	●	●	●	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных	
		MELSEC	IEC
s	Первый адрес операнда, в котором сохранена строковая величина конвертируемых двоичных данных.	строковая величина	массив [1..5]/[1..7] данных типа ANY16
d1	Первый адрес операнда, в котором сохранено количество разрядов двоичных данных после конвертации.	BIN, 16 бит	ANY32
d2	Начальный адрес операнда, в котором сохраняются конвертированные двоичные данные.	BIN, 16/32 бита	ANY16/32

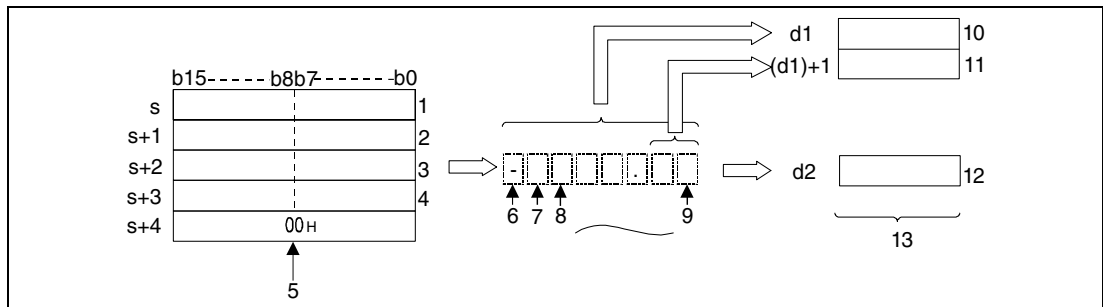
**Принцип действия**

**Конвертирование строковых величин в 16-битные или 32-битные двоичные данные**

**VAL Преобразование в 16-битные двоичные данные**

Команда VAL преобразует строковые величины, хранящиеся в области от s (Array\_s[1]) до s + 4 (Array\_s[5]), в 16-битные двоичные данные. Количество разрядов и двоичное значение сохраняются в d1, (d1) + 1 и d2.

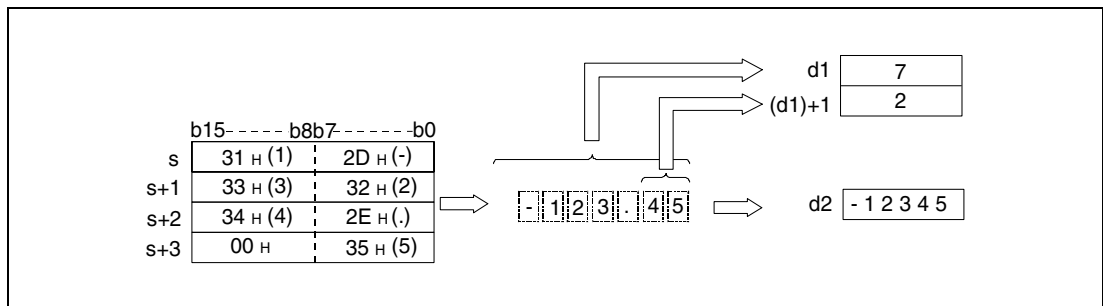
Для конвертирования в 16-битный двоичный формат все данные, находящиеся в области от s (Array\_s[1]) до s + 4 (Array\_s[5]) до знака с кодом "00н", интерпретируются в качестве строковой величины.



- 1 ASCII-код 1-го знака/ASCII-код арифметического знака
- 2 ASCII-код 3-го знака/ASCII-код 2-го знака
- 3 ASCII-код 5-го знака/ASCII-код 4-го знака
- 4 ASCII-код 7-го знака/ASCII-код 6-го знака
- 5 обозначает конец строковой величины
- 6 арифметический знак
- 7 1-й разряд
- 8 2-й разряд
- 9 7-й разряд
- 10 общее количество разрядов
- 11 количество разрядов дробной части
- 12 число типа Integer (десятичная запятая не обрабатывается)
- 13 16-битные двоичные данные

Преобразуется строковая величина "-123.45", хранящаяся в операндах от s (Array\_s[1]) до s + 4 (Array\_s[5]).

Результат сохраняется в d1, (d1) + 1 и d2 следующим образом.



Общее количество знаков, хранящихся в операндах от s (Array\_s[1]) до s + 4 (Array\_s[5]), может находиться в диапазоне между 2 и 8.

Число знаков дробной части, указанное в операндах от  $s$  (Array\_s[1]) до  $s + 4$  (Array\_s[5]), может находиться в диапазоне между 0 и 5. Количество разрядов дробной части не должно превышать общее количество разрядов минус 3.

Числовое значение конвертируемой строковой величины при игнорированной десятичной запятой должно находиться в диапазоне между  $-32768$  и  $32767$ . ASCII-код каждого знака строковой величины, кроме арифметического знака и десятичной запятой, должен находиться в диапазоне между "30н" и "39н".

В качестве положительного арифметического знака используется знак ASCII "20н" (знак пробела).

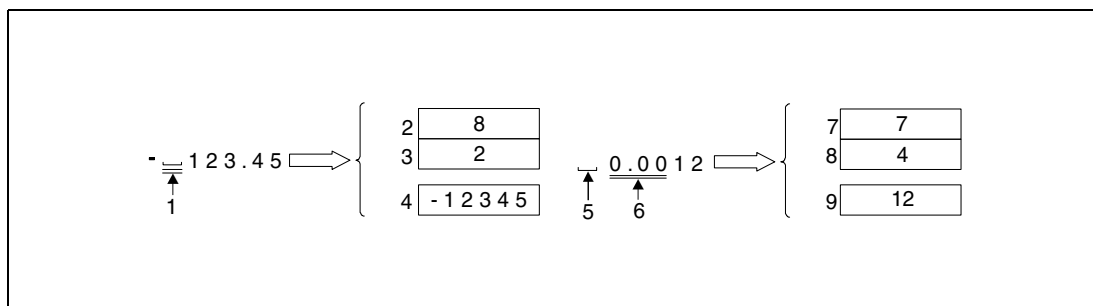
В качестве отрицательного арифметического знака используется знак ASCII "2Dн" (знак минус).

В качестве десятичной запятой используется знак ASCII "2Eн" (точка).

Количество знаков, сохраняемых в  $d1$ ,  $(d1) + 1$  и  $d2$ , образуется из знаков, отображающих числовое значение, арифметического знака  $d1$  и количества знаков дробной части  $(d1) + 1$ .

В двоичных данных, сохраняемых в  $d2$  после конвертации, десятичная запятая отсутствует.

Если между арифметическим знаком и первым значащим разрядом находится знак пробела "20н" или знак нуля "30н", то во время конвертации эти знаки игнорируются.



<sup>1</sup> эти знаки при обработке не учитываются

<sup>2</sup> общее количество разрядов

<sup>3</sup> количество разрядов дробной части

<sup>4</sup> двоичное значение

<sup>5</sup> арифметический знак

<sup>6</sup> эти знаки при обработке не учитываются

<sup>7</sup> общее количество разрядов

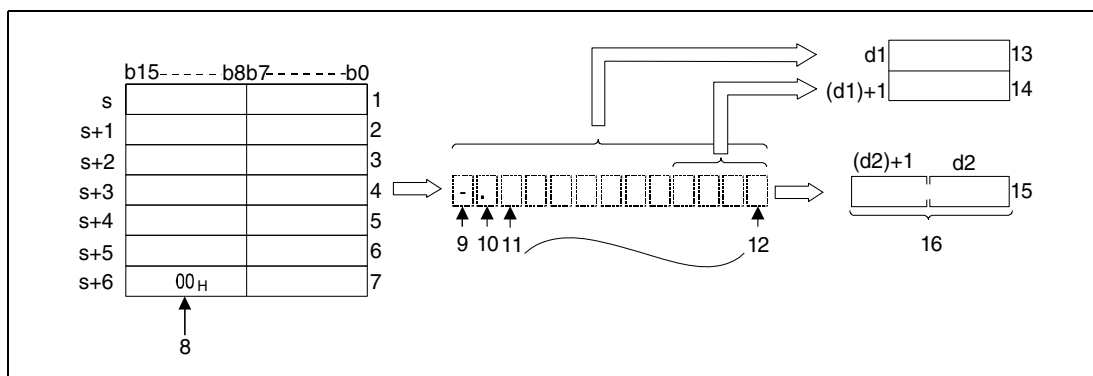
<sup>8</sup> количество разрядов дробной части

<sup>9</sup> двоичное значение

**DVAL Преобразование в 32-битные двоичные данные**

Команда DVAL преобразует строковые величины, сохраненные в операндах от  $s$  ( $\text{Array}_s[1]$ ) до  $s + 6$  ( $\text{Array}_s[7]$ ), в 32-битные двоичные данные. Количество разрядов и двоичное значение сохраняются в  $d1$ ,  $(d1) + 1$ ,  $d2$  и  $(d2) + 1$ .

Для конвертирования в 32-битный двоичный формат все данные в области  $s$  ( $\text{Array}_s[1]$ ) до  $s + 6$  ( $\text{Array}_s[7]$ ) до знака с кодом "00н" интерпретируются как строковая величина.



<sup>1</sup> ASCII-код 1-го знака/ASCII-код арифметического знака

<sup>2</sup> ASCII-код 3-го знака/ASCII-код 2-го знака

<sup>3</sup> ASCII-код 5-го знака/ASCII-код 4-го знака

<sup>4</sup> ASCII-код 7-го знака/ASCII-код 6-го знака

<sup>5</sup> ASCII-код 9-го знака/ASCII-код 8-го знака

<sup>6</sup> ASCII-код 11-го знака/ASCII-код 10-го знака

<sup>7</sup> нулевой ASCII-код/ASCII-код 12-го знака

<sup>8</sup> обозначает конец строковой величины

<sup>9</sup> арифметический знак

<sup>10</sup> 1-й разряд

<sup>11</sup> 2-й разряд

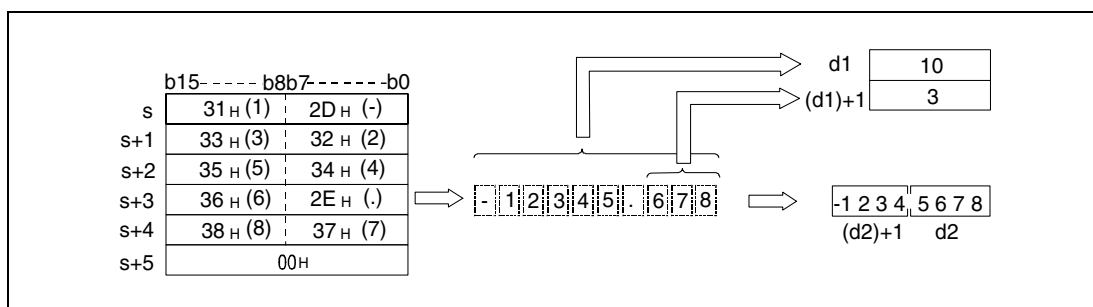
<sup>12</sup> 2-й разряд

<sup>13</sup> общее количество разрядов

<sup>14</sup> количество разрядов дробной части

<sup>15</sup> число типа Integer (десятичная запятая отсутствует)

<sup>16</sup> 32-битное двоичное значение



Общее количество знаков, сохраненных в  $s$  ( $\text{Array}_s[1]$ ) до  $s + 6$  ( $\text{Array}_s[7]$ ), может быть в диапазоне между 2 и 13.

Количество знаков дробной части, сохраненных в области от  $s$  ( $\text{Array}_s[1]$ ) до  $s + 6$  ( $\text{Array}_s[7]$ ), может быть в диапазоне между 0 и 10. Количество разрядов дробной части не должно превышать общее количество разрядов минус 3.

Числовое значение конвертируемой строковой величины при отброшенной десятичной запятой должно находиться в диапазоне между  $-2147483648$  и  $2147483647$ . ASCII-код каждого знака строковой величины, кроме арифметического знака и десятичной запятой, должен находиться в диапазоне между "30н" и "39н".

В качестве положительного арифметического знака используется знак ASCII "20н" (знак пробела).

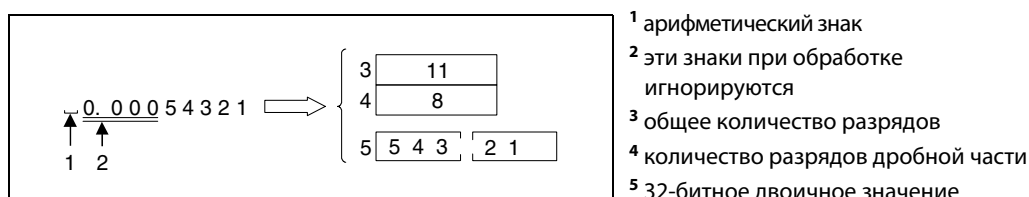
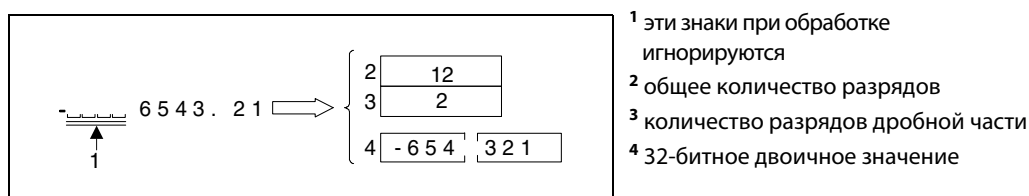
В качестве отрицательного арифметического знака используется знак ASCII "2Dн" (знак минус).

В качестве десятичной запятой вставляется знак ASCII "2Eн" (точка).

Общее количество разрядов, сохраненных в  $d1$ ,  $(d1) + 1$ ,  $d2$  и  $(d2) + 1$ , образуется из знаков, отображающих числовое значение, арифметического знака  $d1$  и числа, обозначающего количество знаков дробной части  $(d1) + 1$ .

В двоичных данных, сохраняемых в  $d2$  и  $(d2) + 1$  после конвертации, десятичная запятая отсутствует.

Если между арифметическим знаком и первым значащим разрядом находится знак пробела "20н" или ноль "30н", то эти знаки во время конвертации пропускаются.



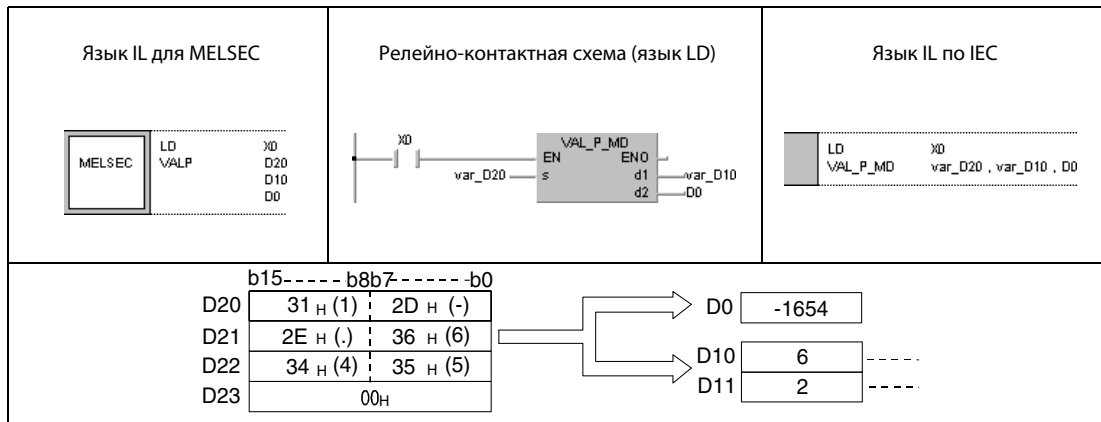
### Источники ошибок

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Общее количество разрядов, сохраненное начиная с  $s$  ( $\text{Array}_s[0]$ ), находится вне диапазона от 2 до 8 (VAL) или от 2 до 13 (DVAL) (код ошибки 4101).
- Количество разрядов дробной части, сохраненное в  $(d1) + 1$ , находится вне диапазона от 0 до 5 (VAL) или 0 до 10 (DVAL) (код ошибки 4100).
- Общее количество разрядов, сохраненное в  $d1$  и  $(d1) + 1$ , превышает количество разрядов дробной части на 3 или более (код ошибки 4100).
- Для арифметического знака применены иные коды ASCII кроме "20н" или "2Dн" (код ошибки 4100).
- В числе встречаются иные коды ASCII кроме "30н", "39н" или "2Eн" (код ошибки 4100).
- В числе имеется более одной десятичной запятой (код ошибки 4100).
- После конвертации двоичное значение находится вне диапазона от  $-32768$  до  $32767$  (VAL) или от  $-2147483648$  до  $2147483647$  (DVAL) (код ошибки 4100).
- Код ASCII "00н" установлен в неправильном месте (код ошибки 4100).

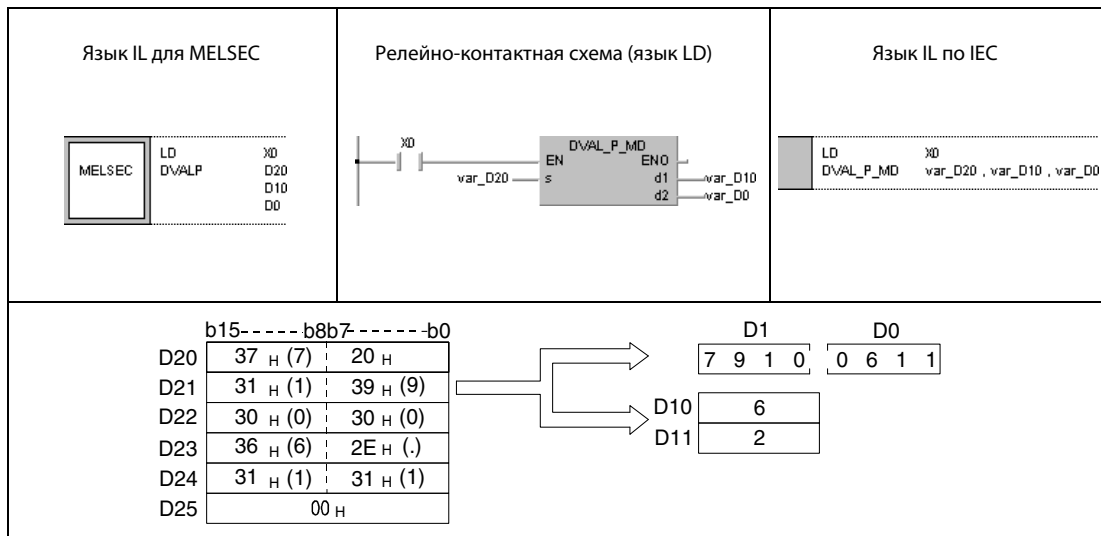
**Пример 1** VALP

Следующая программа при положительном фронте сигнала X0 преобразует строковую величину, хранящуюся в области от D20 (var\_D20 Array [1]) до D23 (var\_D20 Array [4]), в целое число, затем преобразует это число в 16-битный двоичный формат и сохраняет его в D0.



**Пример 2** DVALP

Следующая программа при положительном фронте сигнала X0 преобразует строковую величину, хранящуюся в области от D20 (var\_D20 Array [1]) до D24 (var\_D20 Array [5]), в целое число, затем преобразует это число в 32-битный двоичный формат и сохраняет его в D0 и D1.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке ПОУ эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.



### 7.11.11 ESTR, ESTRP

**Процессор**

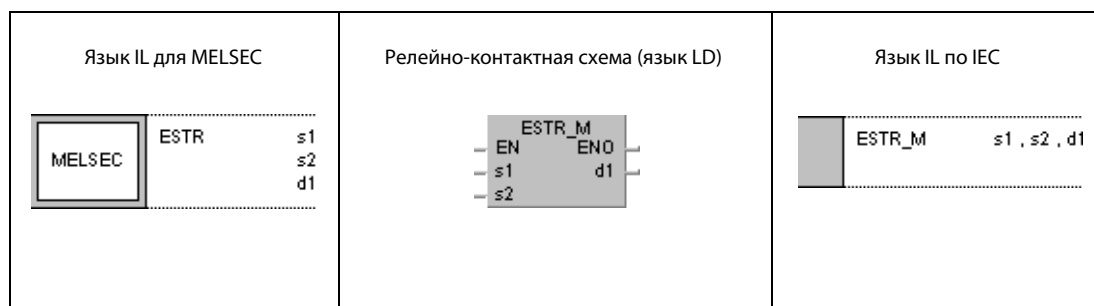
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

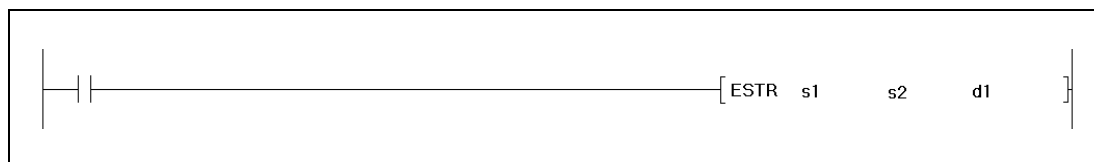
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы E			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	●	●	—	●	—	SMO	4
s2	—	●	●	—	—	—	—	—	—		
d	—	●	●	—	—	—	—	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных	
		MELSEC	IEC
s1	Конvertируемые числа с плавающей запятой или начальный адрес операнда, в котором эти числа сохранены.	REAL	REAL
s2	Первый адрес операндов, в котором сохранен формат воспроизведения конvertируемых числовых данных.	BIN, 16 бит	массив [1..3] данных типа ANY16
d1	Первый адрес операнда, в котором сохраняются конvertированные данные.	строковая величина	строковая величина

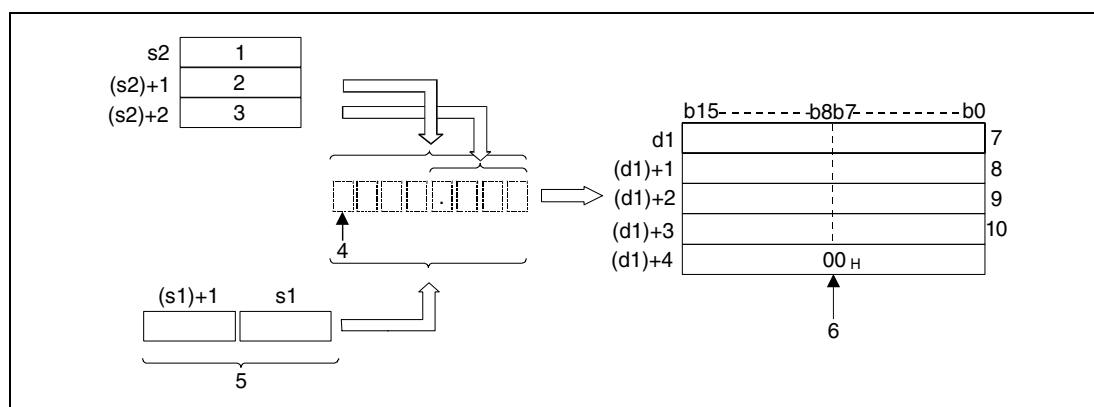
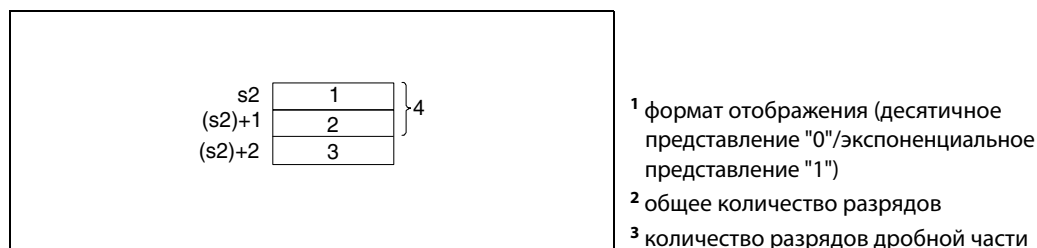
## Принцип действия

## Преобразование чисел с плавающей запятой в строковые величины

## ESTR Преобразование чисел с плавающей запятой

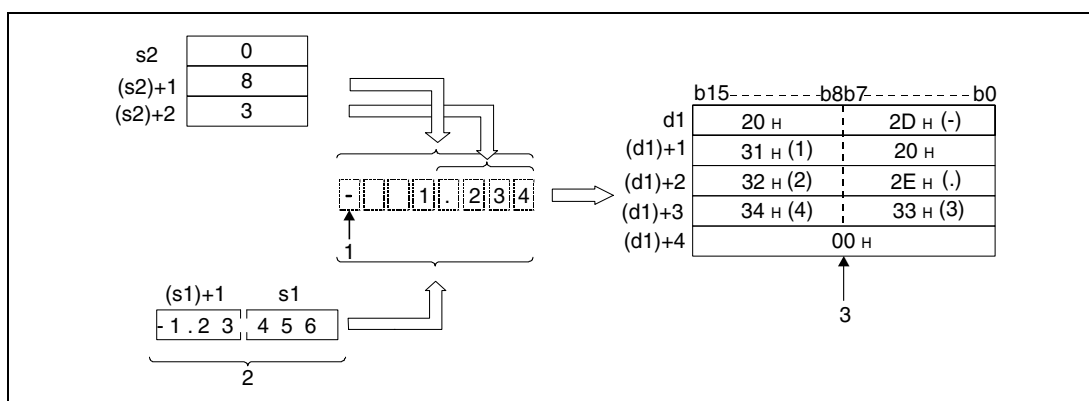
Эта команда преобразует числа с плавающей запятой (данные типа REAL), хранящиеся в  $s1$  и  $(s1) + 1$ , в строковую величину. Формат этой строковой величины указывается в операндах от  $s2$  ( $\text{Array\_s2}[1]$ ) до  $(s2) + 2$  ( $\text{Array\_s2}[3]$ ). Результат сохраняется, начиная с  $d1$ .

Формат отображения данных после конвертации зависит от указания в операндах от  $s2$  ( $\text{Array\_s2}[1]$ ) до  $(s2) + 2$  ( $\text{Array\_s2}[3]$ ).



### Десятичное представление

Число типа REAL –1.23456 преобразуется в строковую величину, содержащую в общей сложности 8 знаков (из них 3 знака являются дробной частью). Результат сохраняется, начиная с d1.



- <sup>1</sup> арифметический знак
- <sup>2</sup> число с плавающей запятой (данные типа REAL)
- <sup>3</sup> автоматически установленный конец строковой величины

Общее количество разрядов конвертируемого числа в (s2) + 1 (Array\_s2[2]) определяется следующим образом:

Если количество разрядов дробной части равно нулю, то общее количество разрядов >= 2.

Если количество разрядов дробной части имеет иное значение, то общее количество разрядов равно 3 + количество разрядов дробной части.

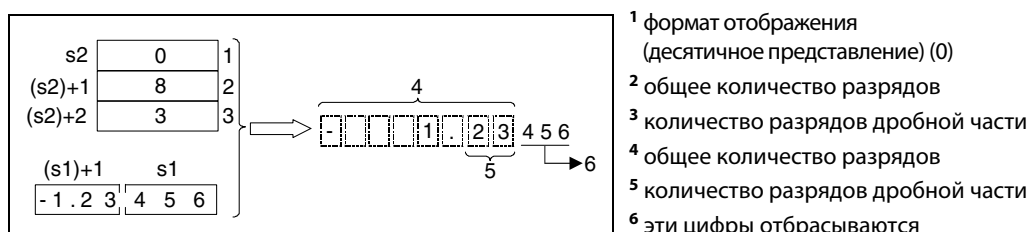
Указываемое количество разрядов дробной части должно находиться в диапазоне между 0 и 7. Количество разрядов дробной части должно быть как минимум на 3 меньше, чем общее количество разрядов.

После преобразования строковая величина сохраняется, начиная с d1, следующим образом:

В качестве положительного арифметического знака числа с плавающей запятой используется знак пробела – ASCII-код "20н".

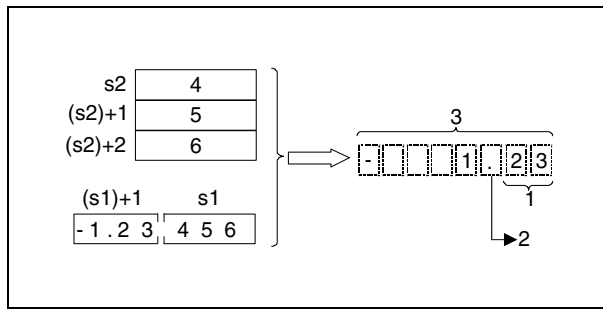
В качестве отрицательного арифметического знака числа с плавающей запятой используется знак минус – ASCII-код "2Dн".

Если фактическое количество разрядов дробной части числа с плавающей запятой превышает введенное количество разрядов дробной части конвертированного числа, то лишние разряды отбрасываются.



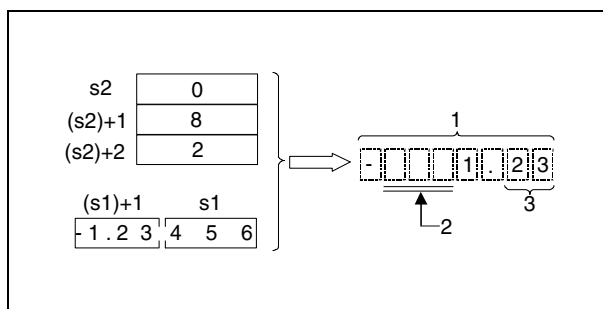
Если количество разрядов дробной части установлено на иное значение кроме нуля, в соответствующем месте автоматически устанавливается десятичная запятая "2Eн" (в виде точки ".").

Если количество разрядов дробной части установлено на ноль, то десятичная запятая "2Eн" (.) не устанавливается.



- 1 количество разрядов дробной части
- 2 десятичная запятая устанавливается и сохраняется автоматически
- 3 общее количество разрядов

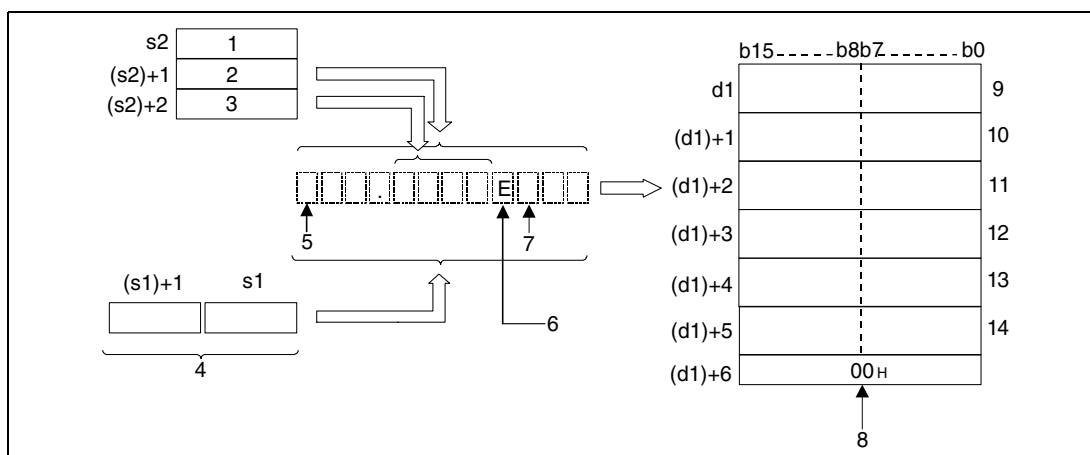
Если общее количество отображаемых разрядов без арифметического знака меньше, чем количество "запятая и дробная часть", то разряды между арифметическим знаком и первым отображаемым разрядом заполняются кодами "20н" (знаками пробела).



- 1 общее количество разрядов
- 2 сохраняются знаки пробела "20н"
- 3 количество разрядов дробной части

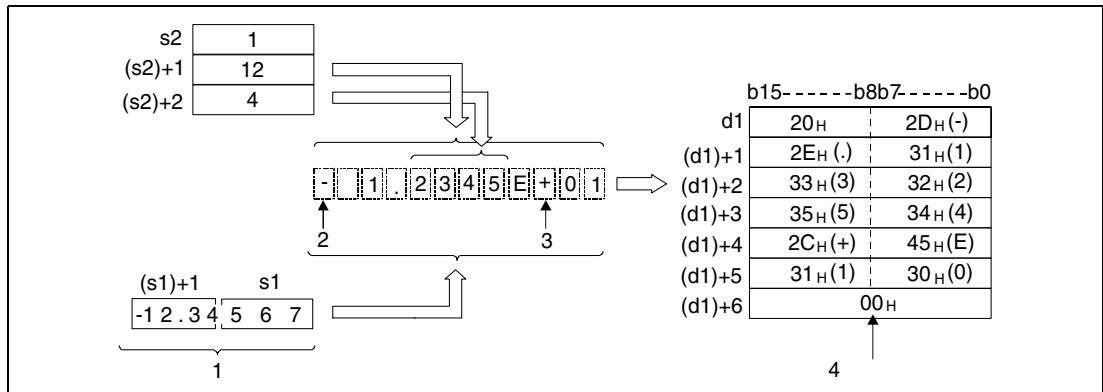
В конце строковой величины автоматически сохраняется код "00н".

## Экспоненциальное представление



- <sup>1</sup> формат отображения (экспоненциальное представление) (1)
- <sup>2</sup> общее количество разрядов
- <sup>3</sup> количество разрядов дробной части
- <sup>4</sup> число с плавающей запятой (данные типа REAL)
- <sup>5</sup> арифметический знак мантииссы
- <sup>6</sup> "E" устанавливается автоматически
- <sup>7</sup> арифметический знак показателя степени
- <sup>8</sup> автоматически установленный конец строковой величины
- <sup>9</sup> ASCII-код знака (общее количество разрядов – 1)/ASCII-код арифметического знака
- <sup>10</sup> ASCII-код знака (общее количество разрядов – 3)/ASCII-код знака (общее количество разрядов – 2)
- <sup>11</sup> ASCII-код знака (общее количество разрядов – 5)/ASCII-код знака (общее количество разрядов – 4)
- <sup>12</sup> ASCII-код знака (общее количество разрядов – 7)/ASCII-код знака (общее количество разрядов – 6)
- <sup>13</sup> арифметический знак показателя степени/45<sub>н</sub> (E)
- <sup>14</sup> ASCII-код знака (общее количество разрядов – 11) (показатель степени)/ASCII-код знака (общее количество разрядов – 10) (показатель степени)

Число типа REAL –12.34567 требуется отобразить в экспоненциальном виде. Общее количество разрядов составляет 12. Из них 4 должны отображать дробную часть десятичного числа. Результат сохраняется, начиная с d1.



- 1 число с плавающей запятой (данные типа REAL)
- 2 арифметический знак мантиссы
- 3 арифметический знак показателя степени
- 4 автоматически установленный конец строковой величины

Общее количество разрядов конвертируемого числа в (s2) + 1 (Array\_s2[2]) определяется следующим образом:

Если количество разрядов дробной части равно нулю, то общее количество разрядов >= 2.

Если количество разрядов дробной части не равно нулю, то общее количество разрядов равно 7 + количество разрядов дробной части.

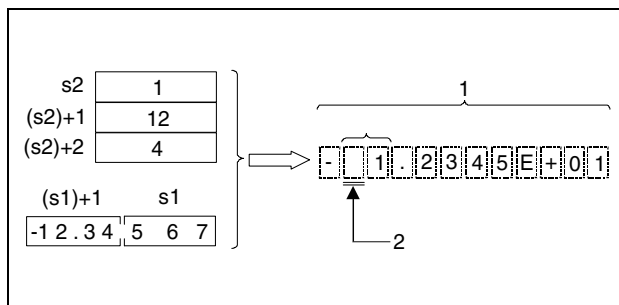
Указываемое количество разрядов дробной части должно находиться в диапазоне между 0 и 7. Количество разрядов дробной части должно быть как минимум на 7 меньше, чем общее количество разрядов.

После преобразования строковая величина сохраняется, начиная с d1, следующим образом:

В качестве положительного арифметического знака десятичной части числа с плавающей запятой используется ASCII-код "20h" (знак пробела).

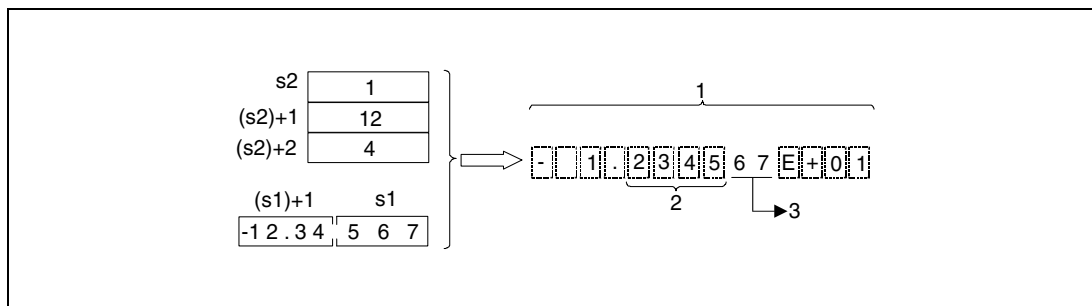
В качестве отрицательного арифметического знака десятичной части числа с плавающей запятой используется ASCII-код "2Dh" (знак минус).

Целочисленная часть установлена на 2 разряда. Если целочисленная часть имеет только один разряд, то между арифметическим знаком и целочисленным разрядом вставляется и сохраняется знак пробела в кодировке ASCII.



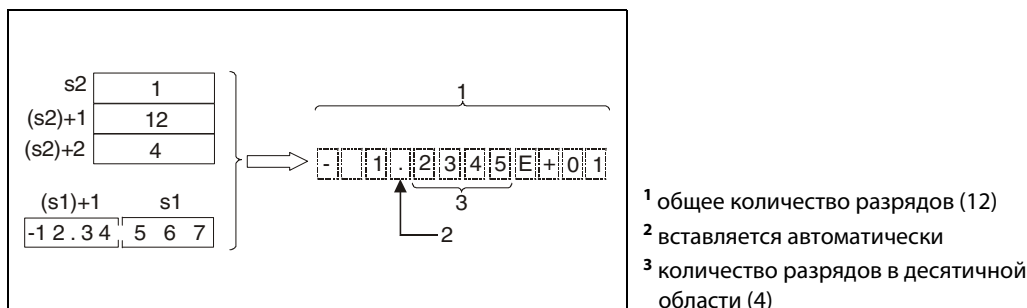
- 1 общее количество разрядов (12)
- 2 записывается знак пробела

Если дробная часть числа с плавающей запятой превышает предусмотренную область памяти, то разряды, которые невозможно сохранить, отбрасываются.



- <sup>1</sup> общее количество разрядов (12)
- <sup>2</sup> количество разрядов в дробной части (4)
- <sup>3</sup> Эти цифры отбрасываются.

Если количество разрядов дробной части установлено на иное значение кроме нуля, то в соответствующем месте автоматически вставляется десятичная запятая "2Ен" (в виде точки ".").

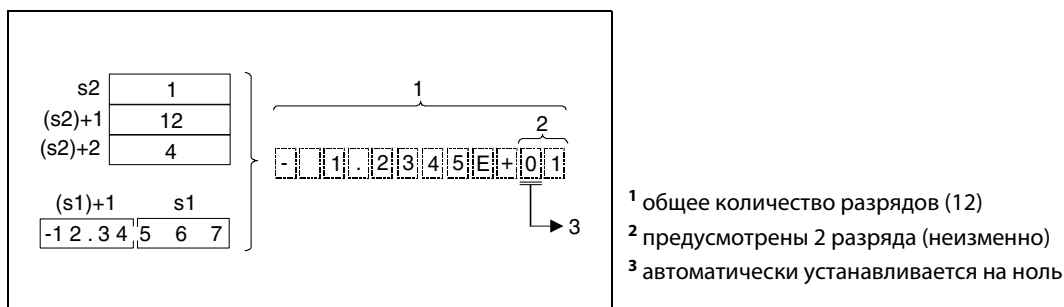


- <sup>1</sup> общее количество разрядов (12)
- <sup>2</sup> вставляется автоматически
- <sup>3</sup> количество разрядов в десятичной области (4)

Если количество разрядов дробной части установлено на ноль, то десятичная запятая "2Ен" не вставляется.

В качестве положительного знака показателя степени устанавливается и сохраняется ASCII-код "2Сн" (знак "+"), а в качестве отрицательного знака – ASCII-код "2Dн" (знак "-").

Для показателя степени выделены 2 разряда. Если в качестве количества разрядов показателя степени указано "1", то между арифметическим знаком показателя степени и показатель степени записывается ASCII-код "30н" (0).



- <sup>1</sup> общее количество разрядов (12)
- <sup>2</sup> предусмотрены 2 разряда (неизменно)
- <sup>3</sup> автоматически устанавливается на ноль

В конце строковой величины автоматически записывается ASCII-код "00н".

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанные в s1 и (s1) + 1 значения не равны нулю и не находятся в пределах диапазонов  $\pm 2^{-127} \leq s1 < \pm 2^{129}$  (код ошибки 4100).
- В качестве формата в s2 (Array\_s2[1]) указано иное число кроме 0 и 1 (код ошибки 4100).
- Общее количество разрядов в (s2) + 1 (Array\_s2[2]) находится вне следующих диапазонов (код ошибки 4100):

*При десятичном формате*

Количество разрядов дробной части равно нулю (при этом общее количество разрядов  $\geq 2$ ).  
Количество разрядов дробной части не равно нулю и при этом общее количество разрядов  $\geq$  (количество разрядов дробной части + 3).

*При экспоненциальном формате*

Количество разрядов дробной части равно нулю (при этом общее количество разрядов  $\geq 2$ ).  
Количество разрядов дробной части не равно нулю и при этом общее количество разрядов  $\geq$  (количество разрядов дробной части + 7).

- Количество разрядов в (s2) + 2 (Array\_s2[3]), образующих дробную часть, находится вне диапазона (код ошибки 4100):

*При десятичном формате*

Количество разрядов, образующих дробную часть, меньше или равно общему количеству разрядов минус 3.

*В экспоненциальном формате*

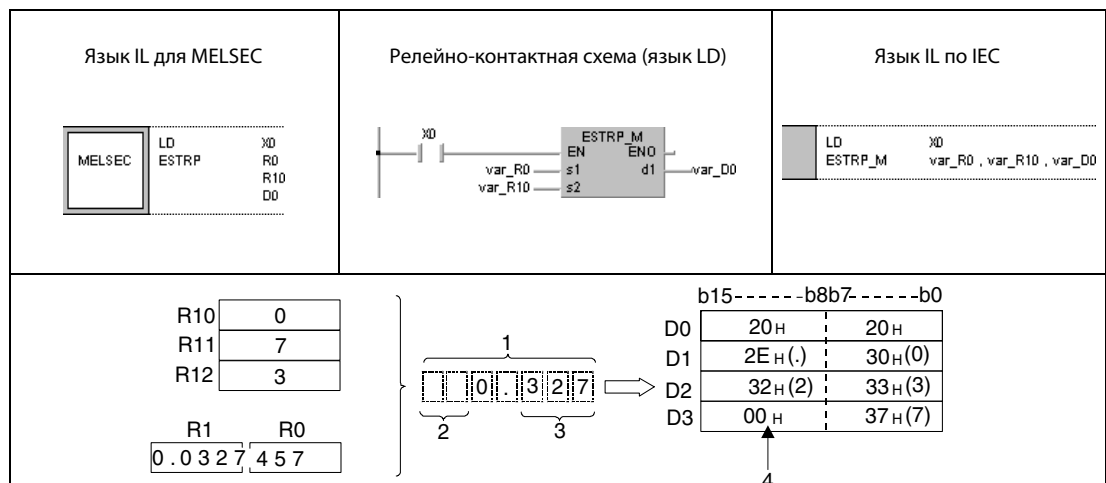
Количество разрядов, образующих дробную часть, меньше или равно общему количеству разрядов минус 7.

- Область памяти начиная с d1 находится вне области операнда, предусмотренной для сохранения (код ошибки 4101).

**Пример 1**

ESTRP

Следующая программа при положительном фронте сигнала X0 преобразует число с плавающей запятой (данные типа REAL), указанное в R0 и R1, в формат, указанный в операндах от R10 (var\_R10 Array [1]) до R12 (var\_R10 Array [3]), и сохраняет результат по адресу D0...D3.

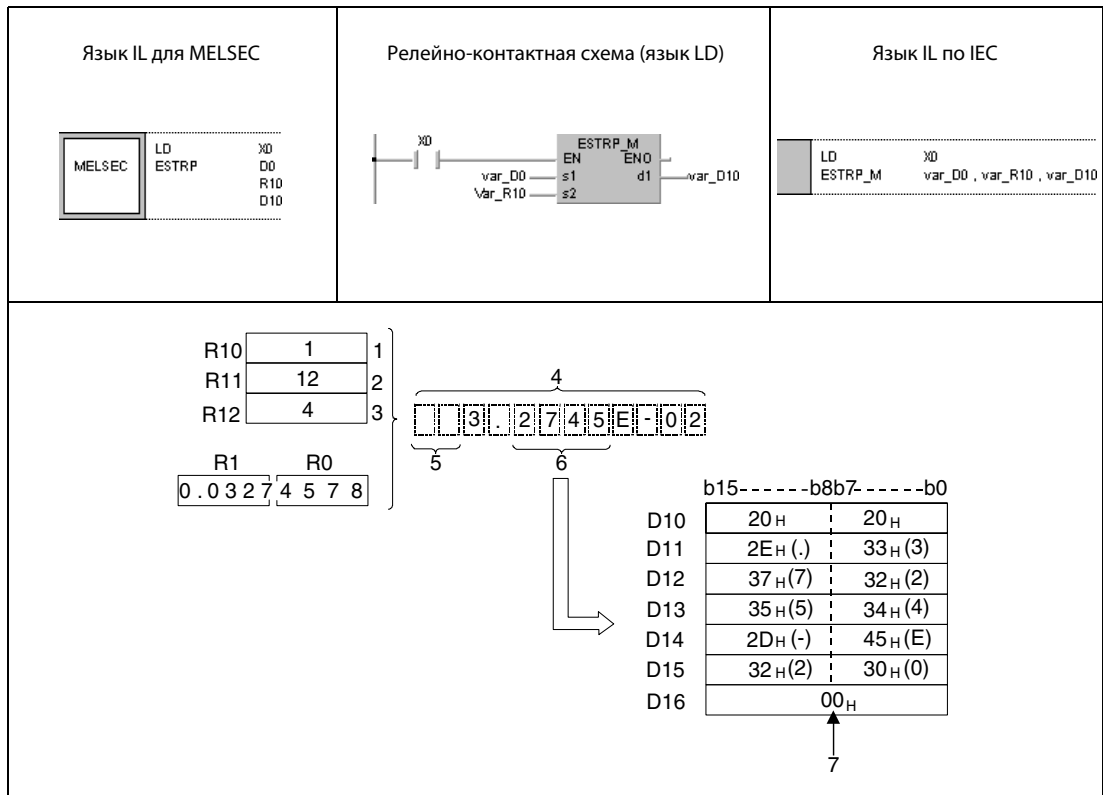


- 1 общее количество разрядов
- 2 знак пробела
- 3 количество разрядов дробной части
- 4 записывается автоматически



**Пример 2** ESTRP

Следующая программа при положительном фронте сигнала X0 преобразует число с плавающей запятой (данные типа REAL), указанное в D0 и D1, в формат, указанный в операндах от R10 (var\_R10 Array [1]) до R12 (var\_R10 Array [3]), и сохраняет результат по адресу D10...D16.



- <sup>1</sup> формат отображения (экспоненциальное представление) (1)
- <sup>2</sup> общее количество разрядов
- <sup>3</sup> количество разрядов дробной части
- <sup>4</sup> общее количество разрядов
- <sup>5</sup> знак пробела
- <sup>6</sup> количество разрядов дробной части
- <sup>7</sup> записывается автоматически

7.11.12 EVAL, EVALP

Процессор

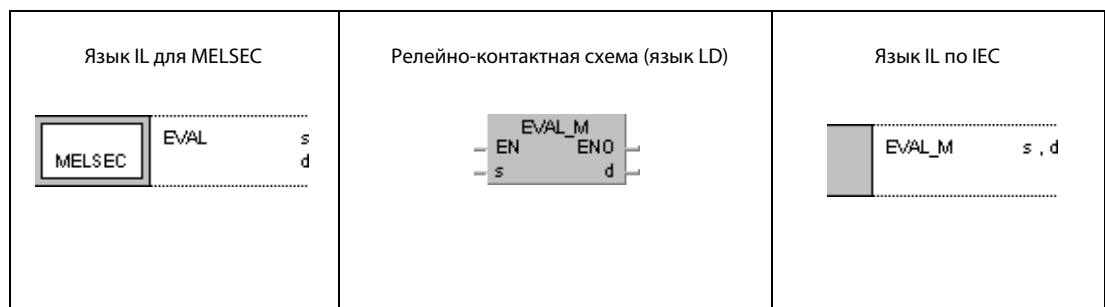
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

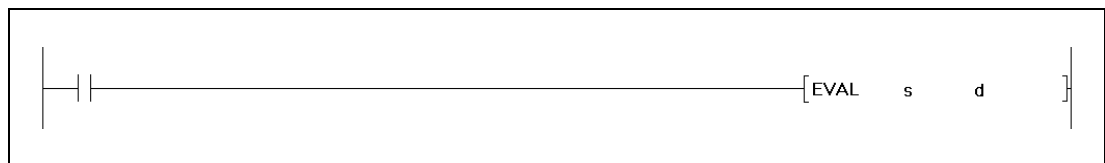
Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□VG□	Индексные регистры Zn	Константы \$			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	●	—	SM0	3
d	—	●	●	—	●	●	—	—	—		

GX IEC  
Developer



GX  
Developer



Переменные

Операнд	Значение	Тип данных
s	Строковая величина, конвертируемая в десятичное число с плавающей запятой (данные типа REAL), или начальный адрес операнда, в котором сохранена эта строковая величина.	строковая величина
d	Первый адрес операнда, в котором после конвертации сохраняется десятичное число с плавающей запятой (данные типа REAL).	REAL

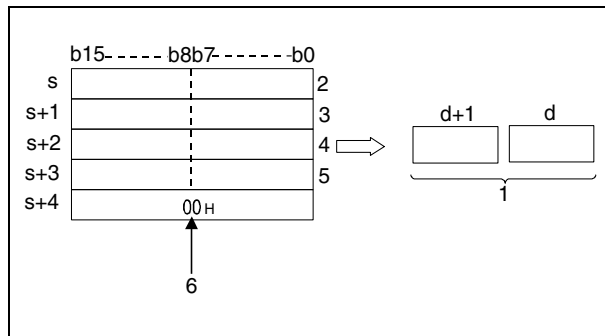
**Принцип действия**

**Преобразование строковых величин в десятичные числа с плавающей запятой**

**EVAL Преобразование строковых величин**

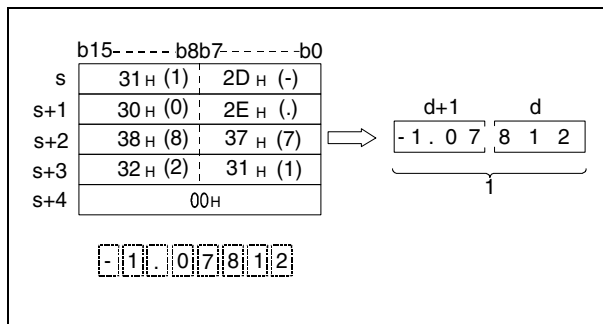
Эта команда преобразует строковую величину, хранящуюся в операндах от s до s + 4, в десятичное число с плавающей запятой (данные типа REAL). Результат сохраняется, начиная с d.

Конвертируемую строковую величину можно преобразовывать либо в десятичный формат с плавающей запятой, либо в экспоненциальный.



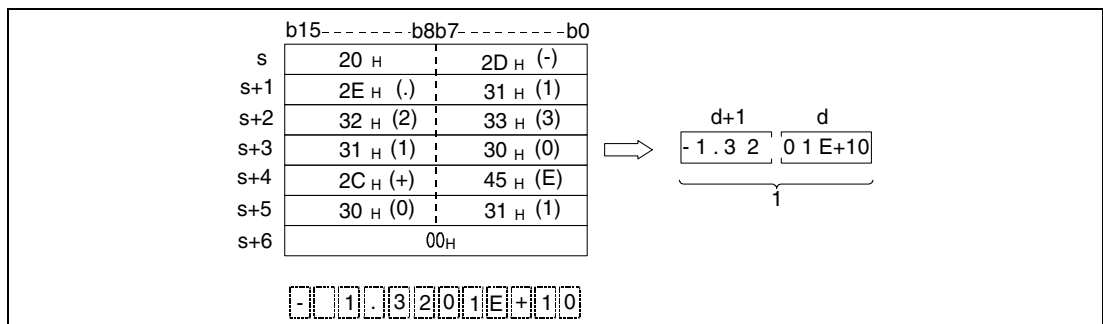
- <sup>1</sup> десятичное число с плавающей запятой (данные типа REAL)
- <sup>2</sup> ASCII-код 1-го знака/ASCII-код арифметического знака
- <sup>3</sup> ASCII-код 3-го знака/ASCII-код 2-го знака
- <sup>4</sup> ASCII-код 5-го знака/ASCII-код 4-го знака
- <sup>5</sup> ASCII-код 7-го знака/ASCII-код 6-го знака
- <sup>6</sup> обозначает конец строковой величины

**Десятичное представление**



- <sup>1</sup> десятичное число с плавающей запятой (данные типа REAL)

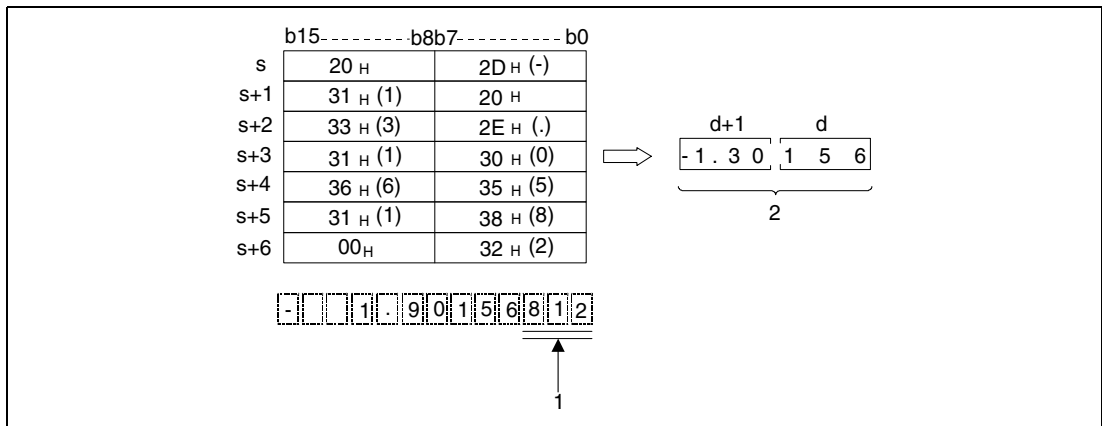
**Экспоненциальное представление**



- <sup>1</sup> десятичное число с плавающей запятой (данные типа REAL)

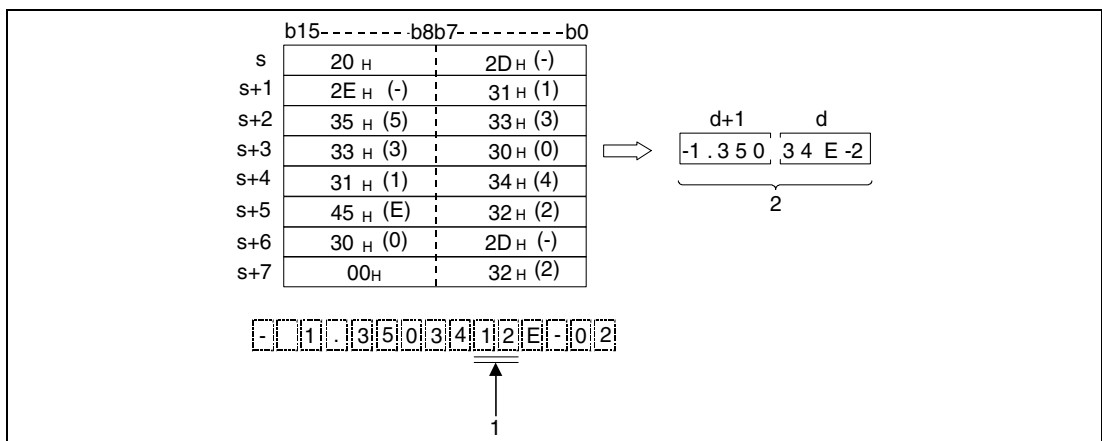
Конвертируются 6 разрядов (без арифметического знака, десятичной запятой и разрядов показателя степени результата) строковой величины, хранящейся, начиная с *s*, в десятичное число с плавающей запятой. Все разряды после 7-го в конвертированном числе отбрасываются.

**Десятичное представление**



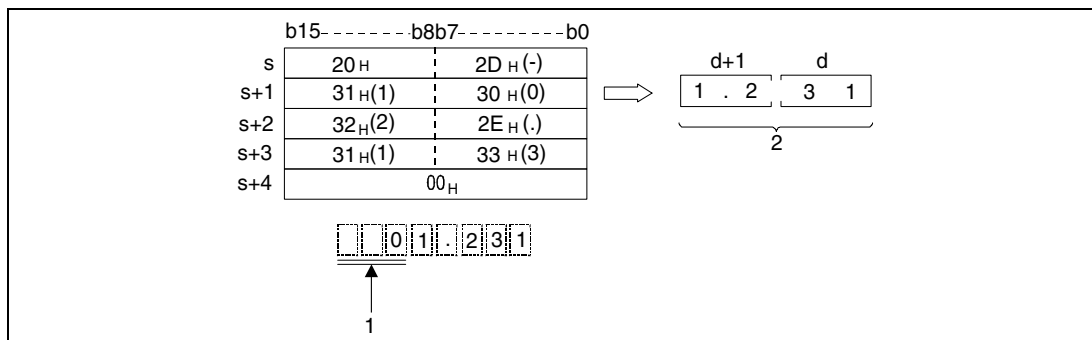
- <sup>1</sup> Эти цифры отбрасываются.
- <sup>2</sup> десятичное число с плавающей запятой (данные типа REAL)

**Экспоненциальное представление**



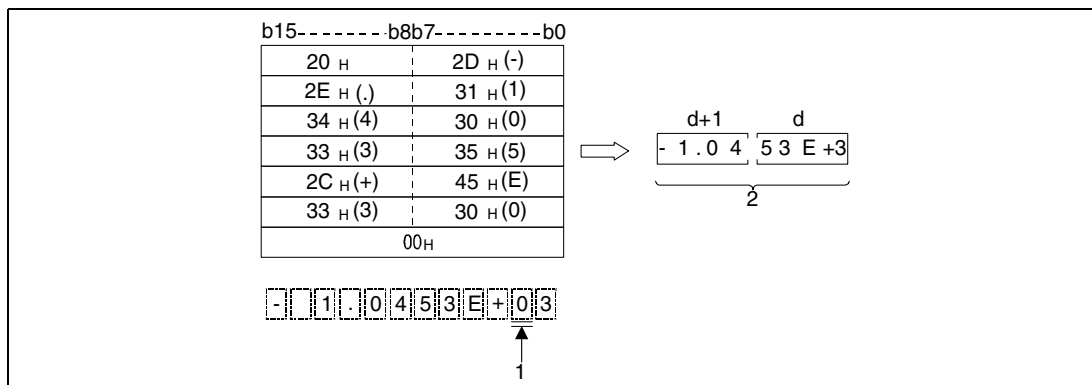
- <sup>1</sup> Эти цифры отбрасываются.
- <sup>2</sup> десятичное число с плавающей запятой (данные типа REAL)

Если в строковой величине, хранящейся, начиная с *s*, перед первой значащей цифрой, имеется ASCII-код пробела "20н" или нуля "30н", то при конвертации эти знаки игнорируются.



- <sup>1</sup> Эти знаки при обработке игнорируются.
- <sup>2</sup> десятичное число с плавающей запятой (данные типа REAL)

Если между знаком "E" и строковой величиной показателя степени имеется ASCII-код нуля "30н", то при конвертации этот знак игнорируется.



- <sup>1</sup> Эта цифра при обработке игнорируется.
- <sup>2</sup> десятичное число с плавающей запятой (данные типа REAL)

Конвертируемая строковая величина может содержать максимум 24 знака.

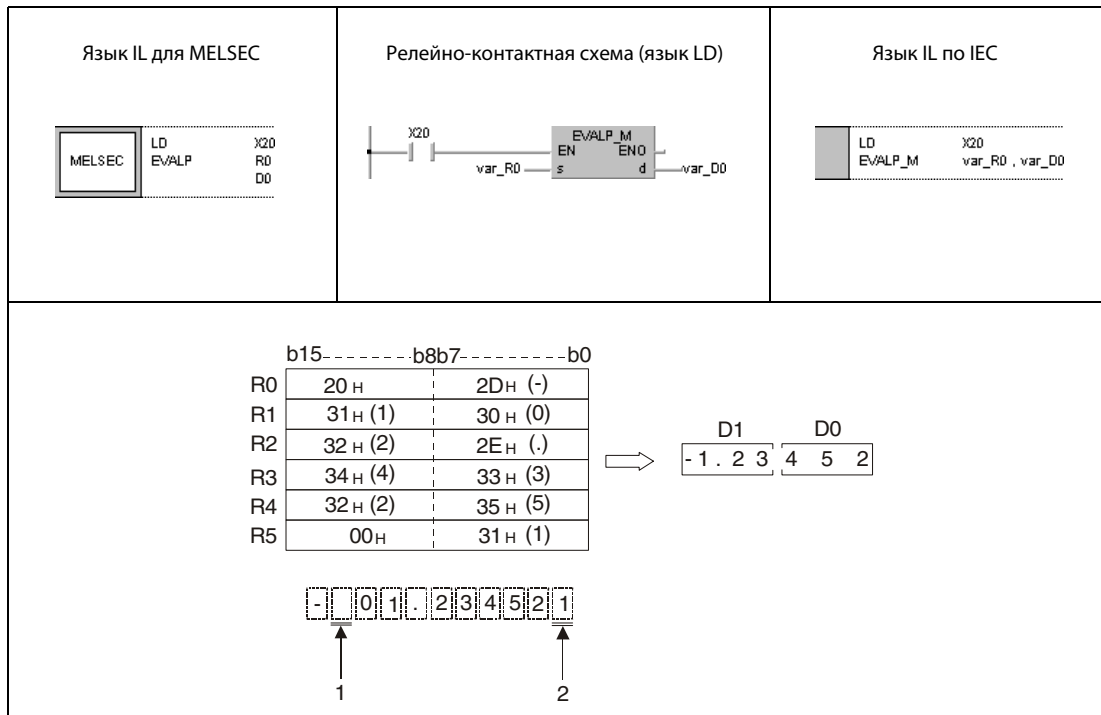
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Строковая величина начинается с иного знака кроме "20н" (знак пробела) или "2Dн" (минус) (код ошибки 4100).
- Перед запятой или дробной частью имеются иные знаки кроме знаков из диапазона от "30н" (0) до "39н" (9) (код ошибки 4100).
- Знак "2Eн" встречается в строковой величине более одного раза (код ошибки 4100).
- В показателе степени использованы иные знаки кроме "45н (E), 2Cн (+)" или "45н (E), 2Dн (-)", или имеется более одного показателя степени (код ошибки 4100).
- Числовое значение не равно нулю и превышает  $1.0 \times 2^{129}$  или меньше  $1.0 \times 2^{-127}$  (код ошибки 4100).
- В области, предусмотренной для сохранения строковой величины, нет метки конца строковой величины "00н" (код ошибки 4100).
- Число знаков в строковой величине равно нулю или превышает 24.

**Пример 1** EVALP

Следующая программа при положительном фронте X20 преобразует строковую величину, указанную в операндах от R0 до R5, в десятичное число с плавающей запятой (данные типа REAL) и сохраняет результат по адресу D0 и D1.

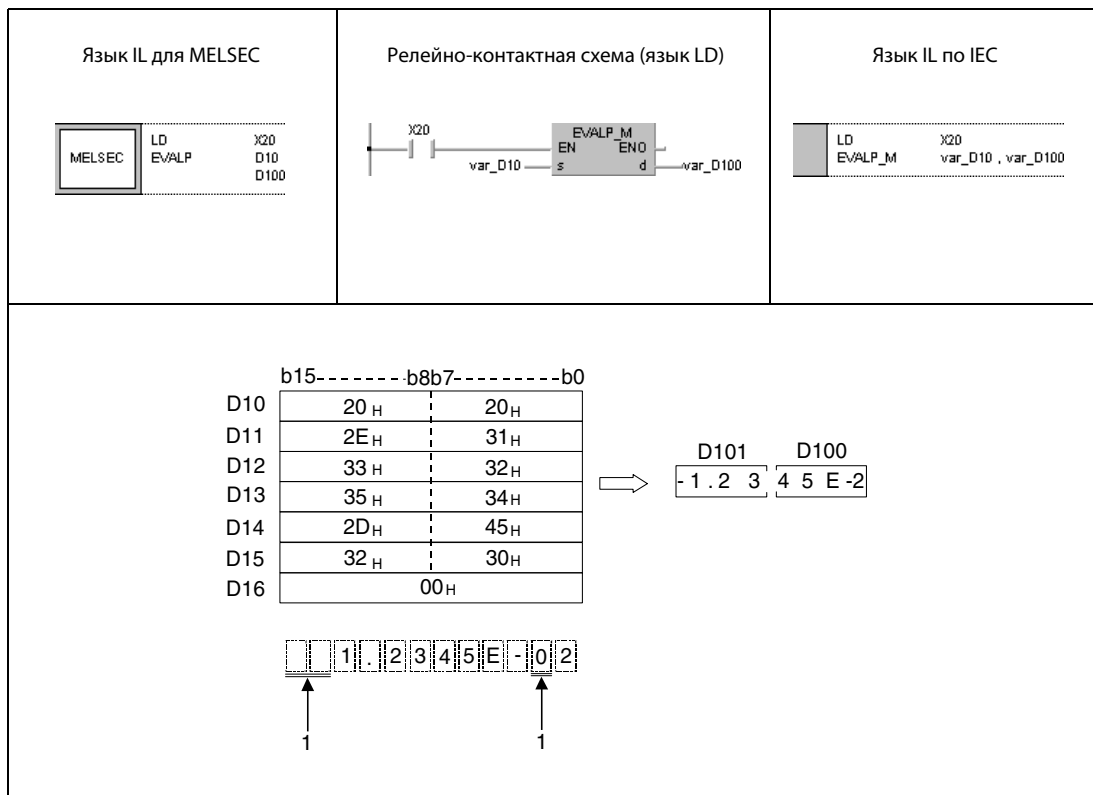


<sup>1</sup> Этот знак при обработке игнорируется.

<sup>2</sup> Эти цифры отбрасываются.

**Пример 2** EVALP

Следующая программа при положительном фронте X20 преобразует строковую величину, указанную в операндах от D10 до D16, в десятичное число с плавающей запятой (данные типа REAL) и сохраняет результат по адресу D100 и D101.



<sup>1</sup> Эти разряды при обработке игнорируются.

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке ПОУ эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.11.13 ASC, ASCP (серии "Q" и "System Q")

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

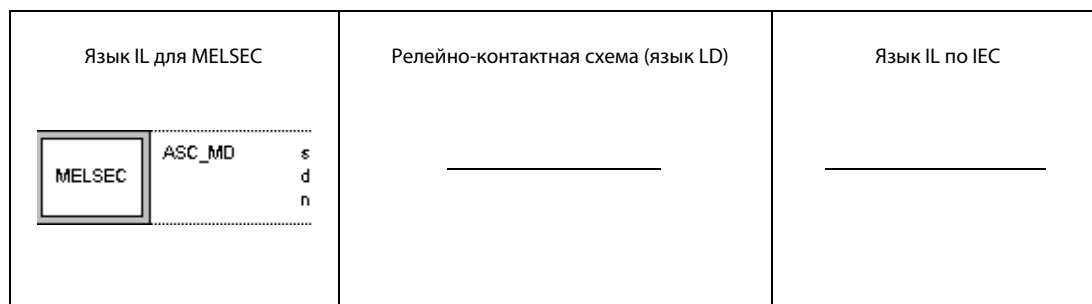
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.ов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□VG□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	SMO	4	
d	—	●	●	—	—	—	—	—			
n	●	●	●	●	●	●	●	●			

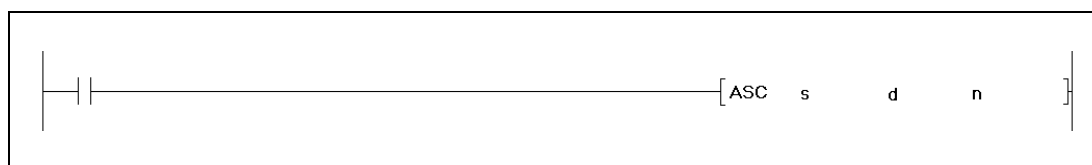
**GX IEC Developer**



В редакторах по стандарту МЭК (IEC) команды ASC и ASCP не действуют. Их можно использовать только при программировании на языке IL для MELSEC.

Выход из положения: непосредственно помещайте в слова данных шестнадцатеричные значения в кодировке ASC.

**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранены конвертируемые двоичные данные.	BIN, 16 бит
n	Количество конвертируемых знаков.	
d	Первый адрес операнда, в котором сохраняются конвертированные строковые величины.	строковая величина

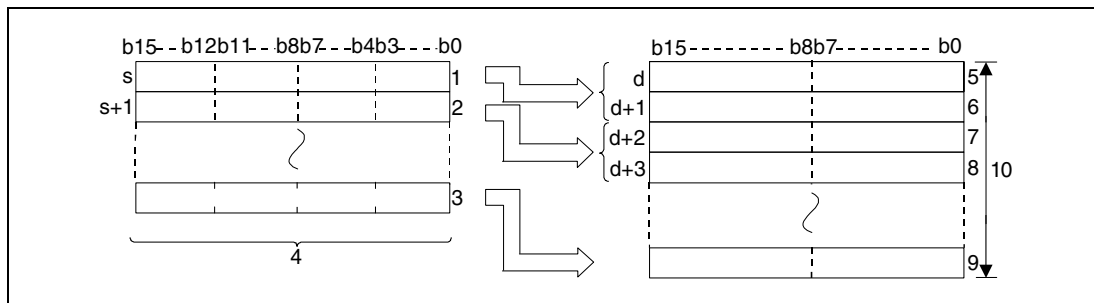


**Принцип действия**

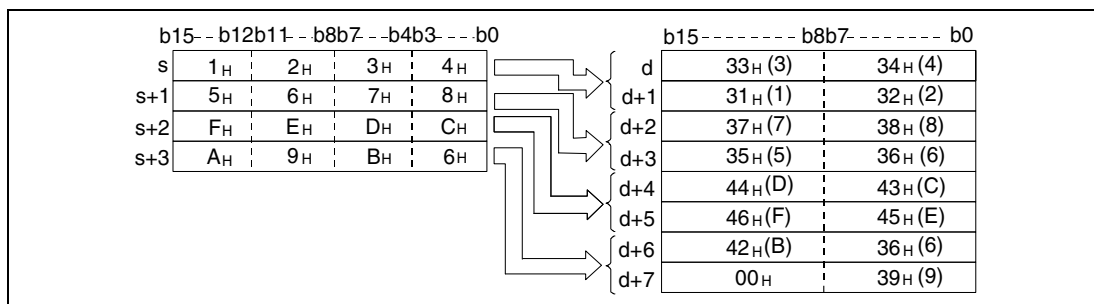
**Преобразование 16-битных двоичных данных в коды ASCII**

**ASC/ASCP Команда конвертации**

Команда ASC преобразует 16-битные двоичные данные, хранящиеся, начиная с s, в шестнадцатеричный формат ASCII и сохраняет результат с указанным в n количеством знаков, начиная с d.

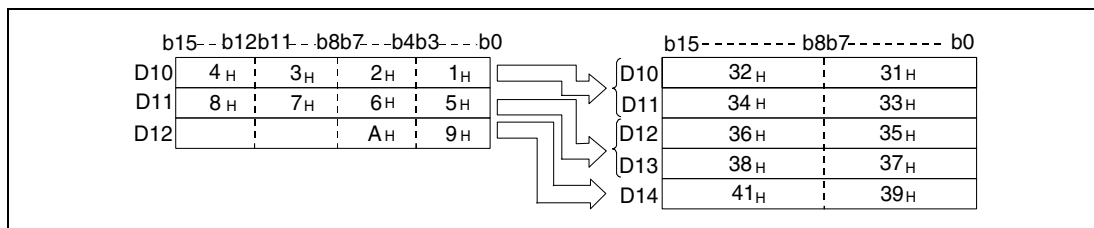


- <sup>1</sup> первый разряд/второй разряд/третий разряд/четвертый разряд
- <sup>2</sup> первый разряд/второй разряд/третий разряд/четвертый разряд
- <sup>3</sup> первый разряд/второй разряд/третий разряд/четвертый разряд
- <sup>4</sup> двоичные данные
- <sup>5</sup> ASCII-код 1-го разряда/ASCII-код 2-го разряда
- <sup>6</sup> ASCII-код 3-го разряда/ASCII-код 4-го разряда
- <sup>7</sup> ASCII-код 5-го разряда/ASCII-код 6-го разряда
- <sup>8</sup> ASCII-код 7-го разряда/ASCII-код 8-го разряда
- <sup>9</sup> ASCII-код 9-го разряда/ASCII-код 10-го разряда
- <sup>10</sup> Количество знаков, указанное в n.

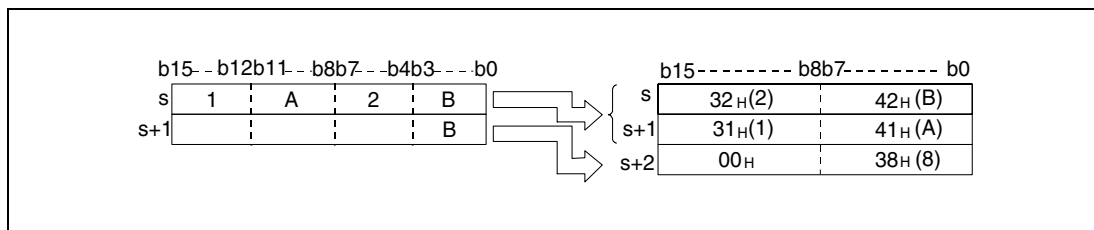


Указанное в n количество знаков определяет диапазоны операндов, начиная с указанных в s и d. Операнды, указанные в s, содержат конвертируемые двоичные данные. В операндах, указанных в d, сохраняется конвертированная строковая величина.

Программа выполняется точно и без сообщений об ошибках даже в том случае, если область, в которой хранятся конвертируемые двоичные данные, перекрывается с областью памяти для конвертированных данных в формате ASCII.



Если в n указано нечетное количество знаков, то в старшие 8 битов последнего адреса области, в которой сохраняется строковая величина, автоматически вставляется ASCII-код "00H".



Если в n (количество знаков) указан ноль, программа не выполняется.

**Источники ошибок**

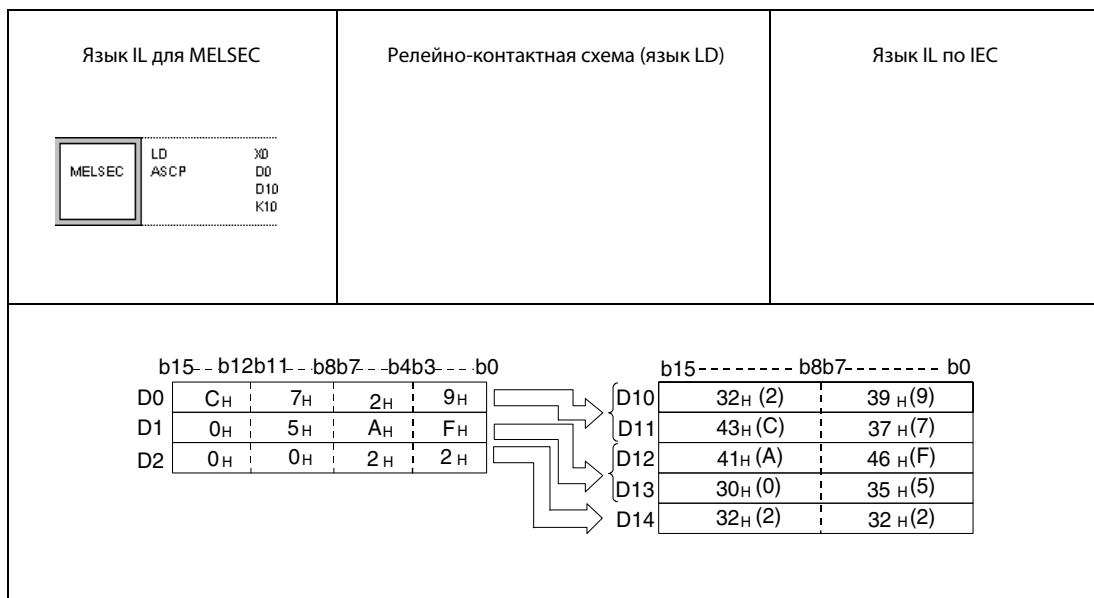
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в n количество знаков (и, тем самым, требуемое количество регистров, начиная с s) находится вне области операнда, предусмотренной для сохранения (код ошибки 4101).
- Указанное в n количество знаков (и, тем самым, требуемое количество регистров, начиная с d) находится вне области операнда, предусмотренной для сохранения (код ошибки 4101).

**Пример**

ASCP

Следующая программа при положительном фронте сигнала X0 считывает хранящиеся в D0 двоичные данные в качестве шестнадцатеричных значений и преобразует их в строковую величину. Результат сохраняется в регистрах с D10 по D14.



### 7.11.14 ASC (серия "A")

**Процессор**

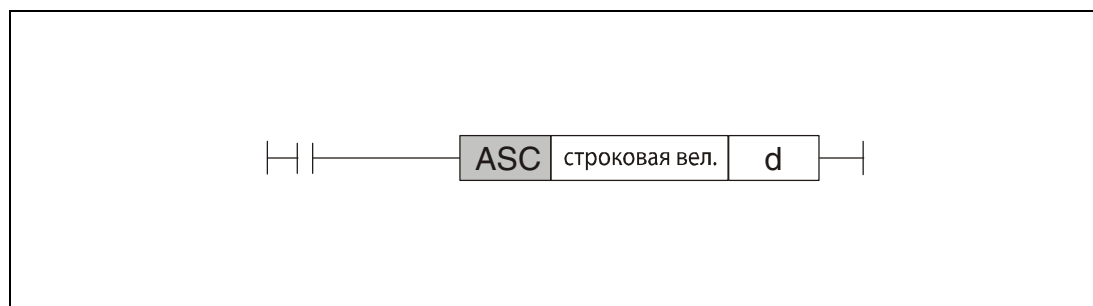
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

**Операнды MELSEC A**

	Операнды													Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки								
	битовые				словные (16 бит)						Константы		Указатели						Уровень							
	X	Y	M	L	S	B	F	T	C	D	W	R	A0						A1	Z	V	K	H (16#)	P	I	N
d								●	●	●	●												13	●		●

<sup>1</sup> Количество шагов в случае применения процессоров AnA, AnU и AnAS указано 3.10.2 „В случае процессоров AnA, AnAS и AnU” этого руководства.

**GX IEC Developer**



**Переменные**

Операнд	Значение	Тип данных
d	Операнд, в котором сохраняются конвертированные знаки.	строковая величина

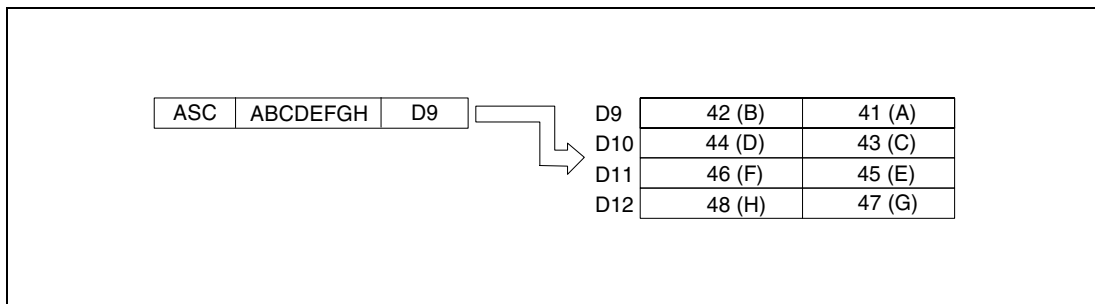
**Принцип действия**

**Преобразование строковых величин в коды ASCII**

**ASC Преобразование буквенно-числовых строковых величин**

Команда ASC преобразует буквенно-числовые строковые величины, содержащие до 8 знаков, в коды ASCII. Результат сохраняется, начиная с d.

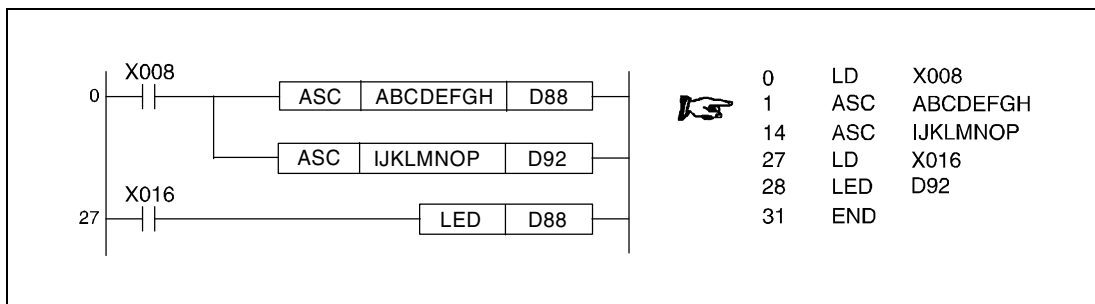
Сохраненные коды ASCII можно распечатать с помощью команд PR/PRC или вывести на светодиодный дисплей модулей центральных процессоров, оснащенных дисплеем, с помощью команды LED.



**Пример**

**ASCP**

Следующая программа после включения входа X8 преобразует строковую величину "ABCDEFGH IJKLMNOP" в коды ASCII и сохраняет результат в регистрах с D88 по D91 и с D92 по D95. После включения входа X16 коды ASCII из регистров с D88 по D95 отображаются на светодиодном дисплее с передней стороны модуля центрального процессора.



### 7.11.15 HEX, HEXP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

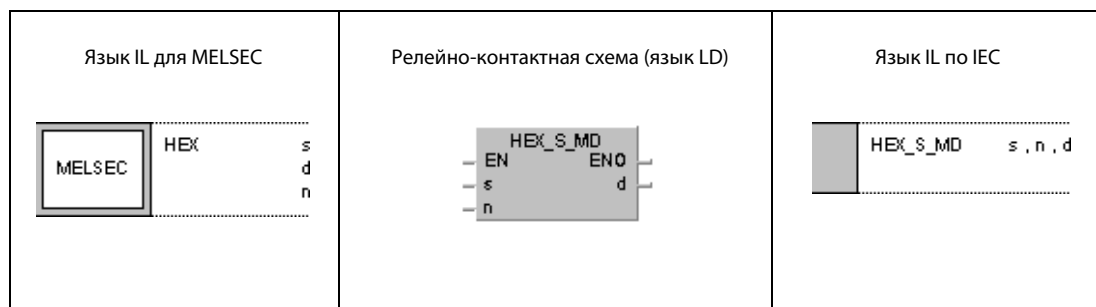
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

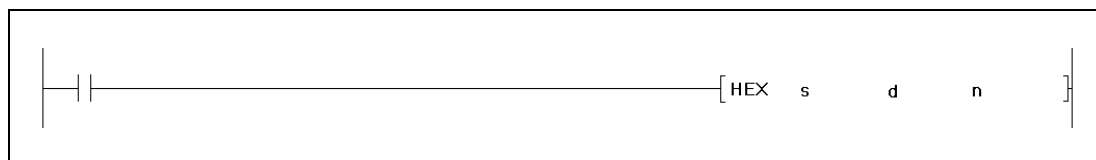
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□\G□□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	SM0	4	
d	—	●	●	—	—	—	—	—			
n	●	●	●	●	●	●	●	●			

**GX IEC Developer**



**GX Developer**



**Переменные**

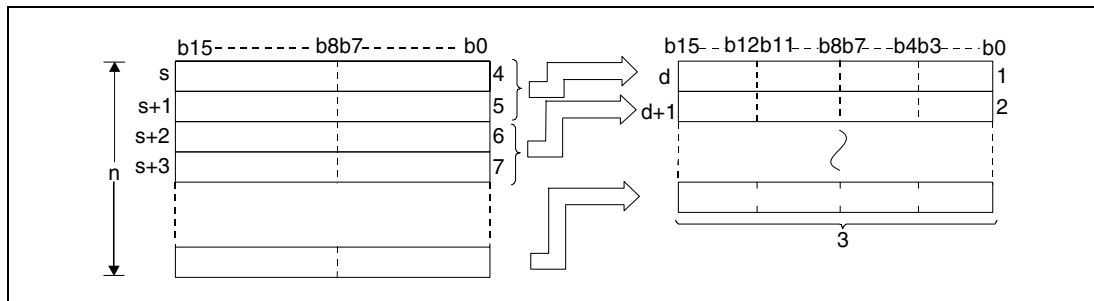
Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранены строковые величины, конвертируемые в двоичные данные.	строковая величина
d	Первый адрес операнда, в котором сохраняются конвертированные двоичные данные.	BIN, 16 бит
n	Количество конвертируемых знаков.	

## Принцип действия

## Преобразование шестнадцатеричных значений ASCII в двоичные значения

## HEX Преобразование шестнадцатеричных значений ASCII

Команда преобразует шестнадцатеричные знаки ASCII, хранящиеся начиная с *s*, в двоичные значения. Количество конвертируемых знаков устанавливается в *n*. Результат конвертации сохраняется, начиная с *d*.



<sup>1</sup> четвертый разряд/третий разряд/второй разряд/первый разряд

<sup>2</sup> четвертый разряд/третий разряд/второй разряд/первый разряд

<sup>3</sup> двоичные данные

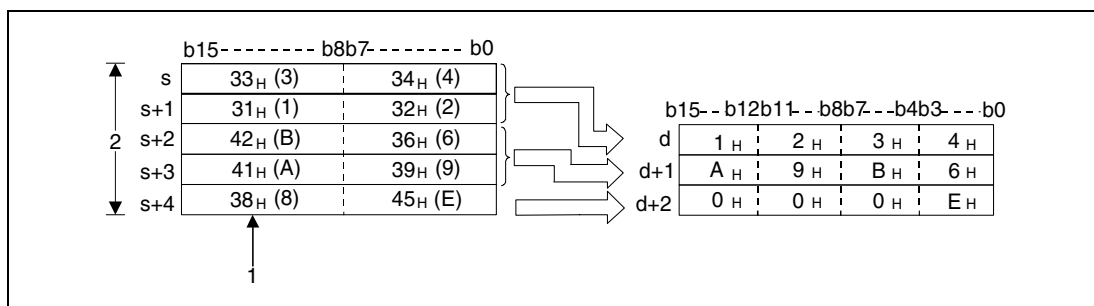
<sup>4</sup> ASCII-код 2-го разряда/ASCII-код 1-го разряда

<sup>5</sup> ASCII-код 4-го разряда/ASCII-код 3-го разряда

<sup>6</sup> ASCII-код 2-го разряда/ASCII-код 1-го разряда

<sup>7</sup> ASCII-код 4-го разряда/ASCII-код 3-го разряда

В *n* указано 9 знаков.

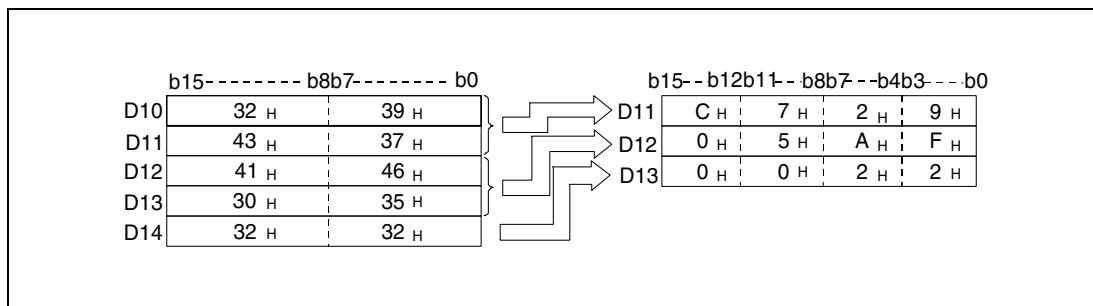


<sup>1</sup> Так как указано, что строковая величина состоит из 9 знаков, код "38н" не изменяется и не передается.

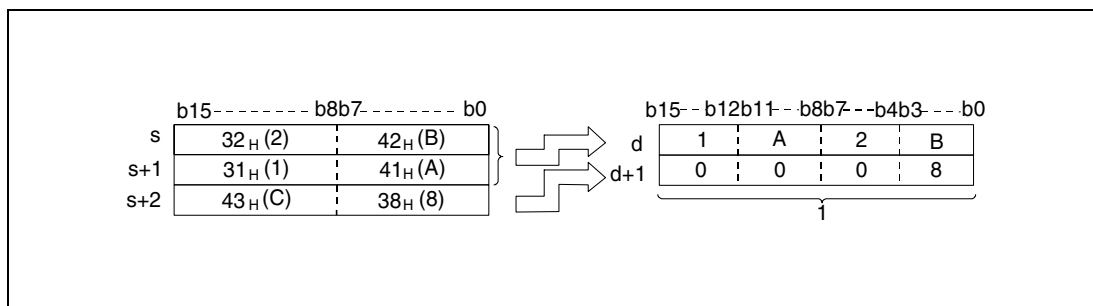
<sup>2</sup> *n* = 9

В результате указания количества знаков в *n* автоматически устанавливается область, занимаемая строковой величиной, начиная с *s*, и область двоичных данных, начиная с *d*.

Даже если область конвертируемых ASCII-кодов перекрывается с областью конвертированных двоичных значений, команда работает правильно.



Если число знаков в n не кратно 4, то после указанного количества знаков в последнем регистре, предусмотренном для конвертированных двоичных значений, автоматически записывается ноль.



<sup>1</sup> Ноль записывается автоматически.

Если число знаков в n равно нулю, процесс конвертации не выполняется.

ASCII-коды, хранящиеся начиная с s, могут находиться в следующих диапазонах: от "30н" до "39н" и от "41н" до "46н".

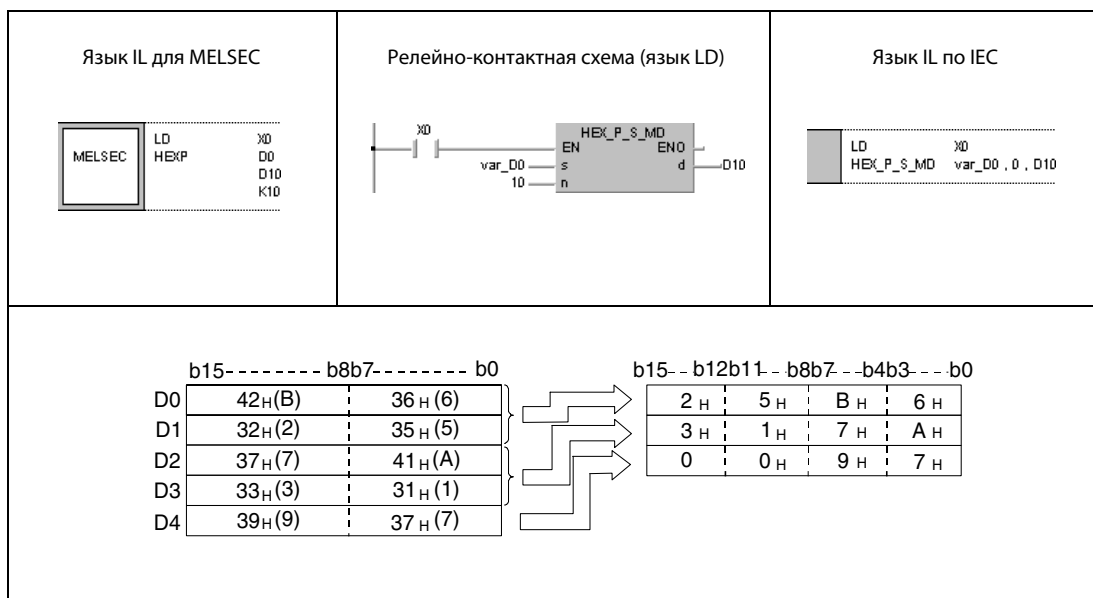
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Операнды начиная с s содержат знаки, находящиеся вне диапазонов от "30н" до "39н" и от "41н" до "46н" (код ошибки 4100).
- Указанное в n количество знаков (и, тем самым, требуемое количество регистров, начиная с s) находится вне области операнда, предусмотренной для сохранения (код ошибки 4101).
- Указанное в n количество знаков (и, тем самым, требуемое количество регистров, начиная с d) находится вне области операнда, предусмотренной для сохранения (код ошибки 4101).
- В n указано отрицательное значение.

**Пример** HEXP

Следующая программа при положительном фронте сигнала X0 преобразует строковую величину "6B52A71379", хранящуюся в регистрах с D0 по D4, в двоичные данные и сохраняет результат в регистрах с D10 по D14.



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.



### 7.11.16 RIGHT, RIGHTP, LEFT, LEFTP

**Процессор**

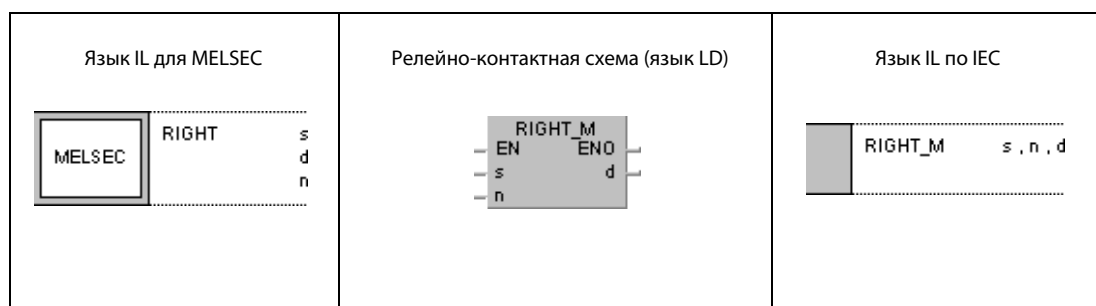
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

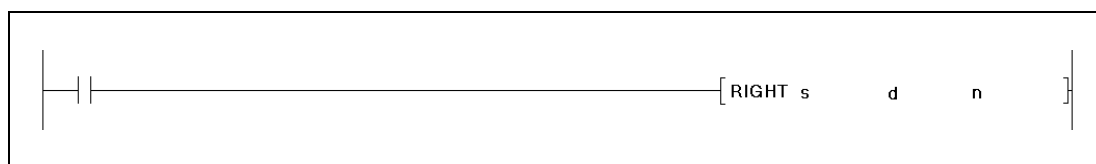
**Операнды MELSEC Q**

	Операнды										Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы		Иные U		
	битовые	словные		битовые	словные			K, H (16#)	\$			
s	—	●	●	—	—	—	—	—	●	—	SMO	4
d	—	●	●	—	—	—	—	—	—	—		
n	●	●	●	●	●	●	●	●	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

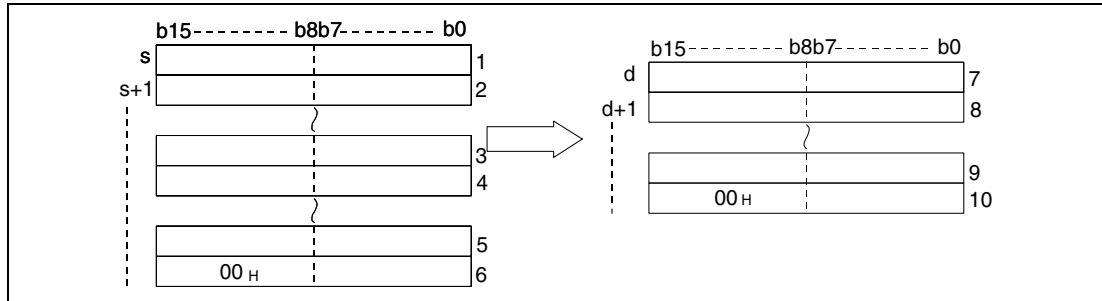
Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранена строковая величина.	строковые величины
d	Первый адрес области операндов, в которой сохраняются считанные знаки строковой величины.	
n	Число считываемых знаков слева или справа	BIN, 16 бит

**Принцип действия**

**Фрагмент строковых данных справа или слева**

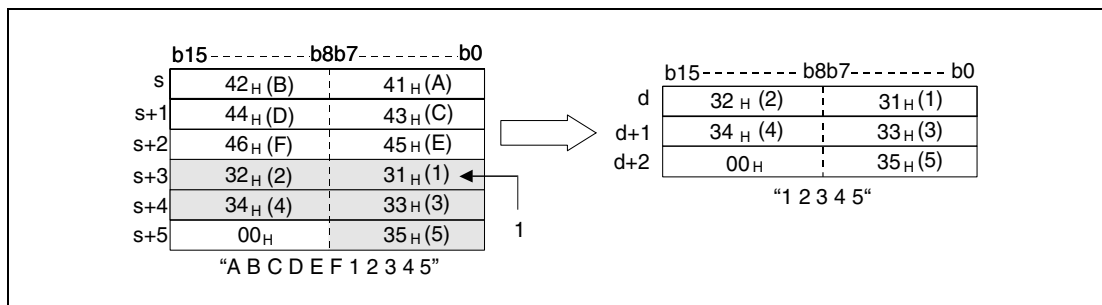
**RIGHT Фрагмент строковых данных справа**

Команда RIGHT считывает n знаков с правой стороны строковой величины (конец строковой величины), хранящейся начиная с s, и сохраняет их, начиная с d.



- <sup>1</sup> ASCII-код 2-го знака/ASCII-код 1-го знака
- <sup>2</sup> ASCII-код 4-го знака/ASCII-код 3-го знака
- <sup>3</sup> ASCII-код знака (последний знак минус n + 2)/ASCII-код знака (последний знак минус n + 1)
- <sup>4</sup> ASCII-код знака (последний знак минус n + 4)/ASCII-код знака (последний знак минус n + 3)
- <sup>5</sup> ASCII-код знака (последний знак минус 1)/ASCII-код знака (последний знак минус 2)
- <sup>6</sup> "00H"/ASCII-код последнего знака
- <sup>7</sup> ASCII-код знака (последний знак минус n + 2)/ASCII-код знака (последний знак минус n + 1)
- <sup>8</sup> ASCII-код знака (последний знак минус n + 4)/ASCII-код знака (последний знак минус n + 3)
- <sup>9</sup> ASCII-код знака (последний знак минус 1)/ASCII-код знака (последний знак минус 2)
- <sup>10</sup> "00H"/ASCII-код последнего знака

при n = 5

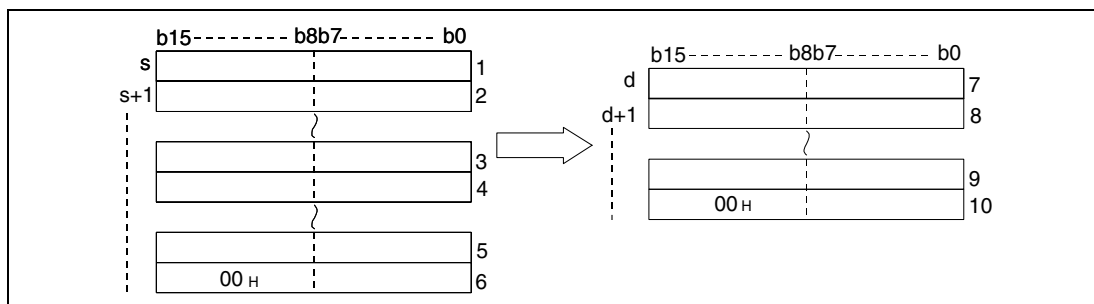


- <sup>1</sup> ASCII-код 5-го знака

Если число знаков в n равно нулю, в d сохраняется код "00H".

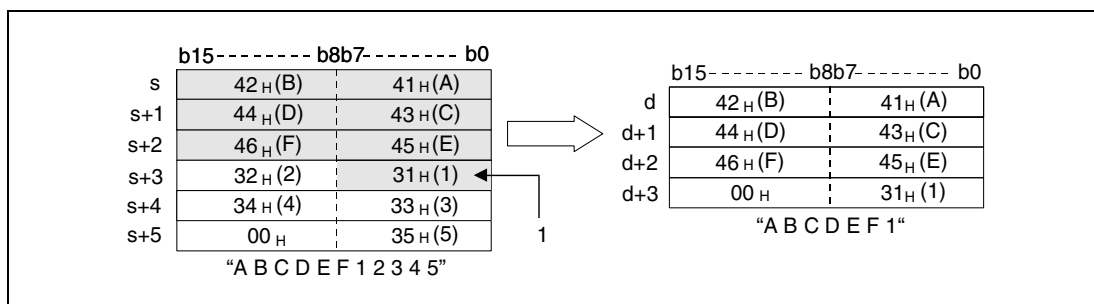
**LEFT**                      **Фрагмент строковых данных слева**

Команда LEFT считывает n знаков с левой стороны строковой величины (начало строковой величины), хранящейся начиная с s, и сохраняет их, начиная с d.



- <sup>1</sup> ASCII-код 2-го знака/ASCII-код 1-го знака
- <sup>2</sup> ASCII-код 4-го знака/ASCII-код 3-го знака
- <sup>3</sup> ASCII-код знака n – 1/ASCII-код знака n – 2
- <sup>4</sup> ASCII-код знака n + 1/ASCII-код n-ного знака
- <sup>5</sup> "00H"/ASCII-код последнего знака
- <sup>6</sup> ASCII-код 2-го знака/ASCII-код 1-го знака
- <sup>7</sup> ASCII-код 4-го знака/ASCII-код 3-го знака
- <sup>8</sup> ASCII-код знака n – 1/ASCII-код знака n – 2
- <sup>9</sup> "00H"/ASCII-код n-ного знака

при n = 7



- <sup>1</sup> ASCII-код 7-го знака

Если число знаков в n равно нулю, в d сохраняется код "00H".

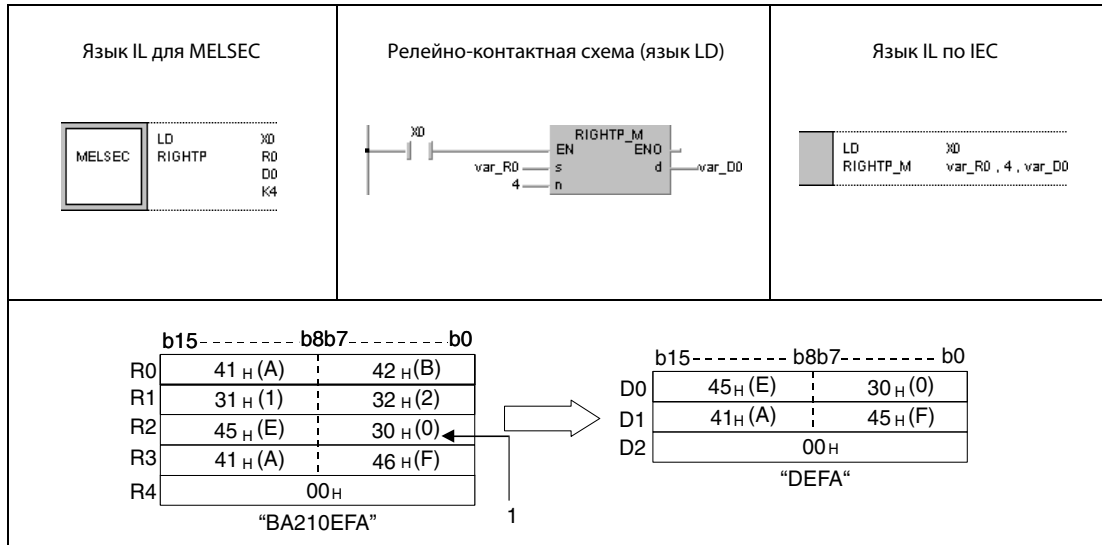
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Значение в n превышает количество знаков, хранящихся, начиная с s (код ошибки 4101).
- Область, указанная с помощью n, находится вне области операнда, предусмотренной для сохранения, начиная с d (код ошибки 4101).

**Пример 1** RIGHTP

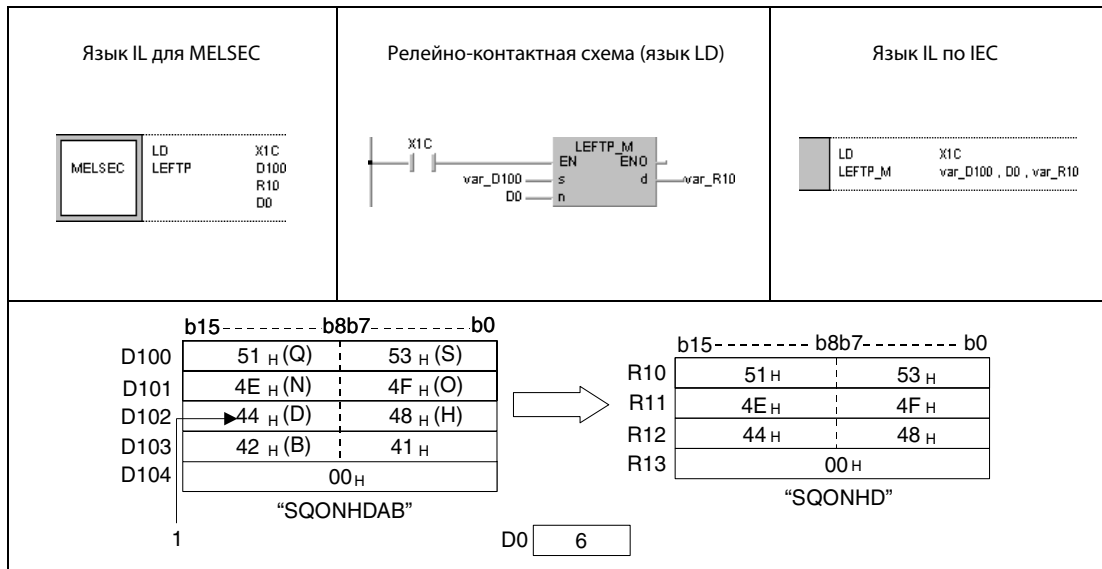
Следующая программа при положительном фронте сигнала X0 определяет 4 правых знака строковой величины, хранящейся в регистрах с R0 по R4, и сохраняет эти знаки в регистрах с D0 по D2.



<sup>1</sup> ASCII-код 4-го знака

**Пример 2** LEFTP

Следующая программа при положительном фронте сигнала X1C определяет указанное в D0 количество знаков с левой стороны строковой величины, указанной в регистрах с D100 по D104, и сохраняет эти знаки в регистрах с R10 по R13.



<sup>1</sup> ASCII-код 6-го знака

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.11.17 MIDR, MIDRP, MIDW, MIDWP

**Процессор**

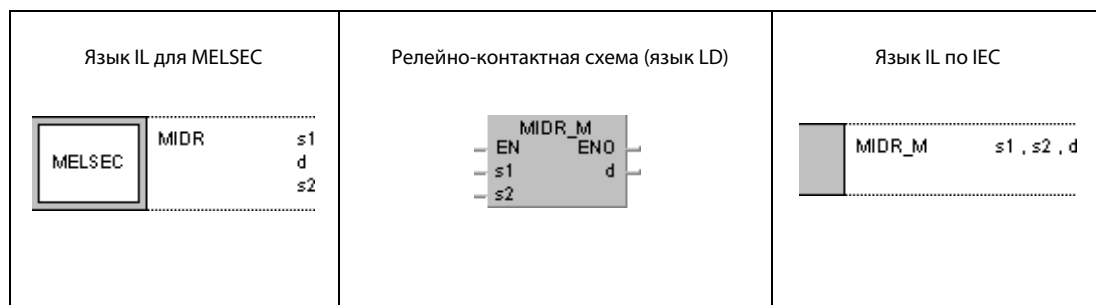
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

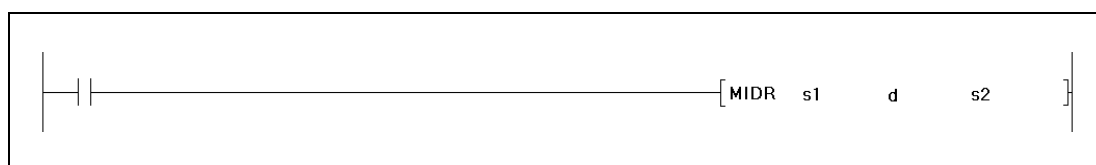
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы \$			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	●	—	SM0	4
d	—	●	●	—	—	—	—	—	—		
s2	●	●	●	●	●	●	●	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных	
		MELSEC	IEC
s1	Первый адрес операнда, в котором сохранена строковая величина.	строковая величина	строковая величина
d	Первый адрес операнда, в котором сохраняется полученная строковая величина.		
s2	Первый адрес операнда, в котором указан номер первого считываемого знака и число знаков. (s2) + 0 (Array_s2[1]): регистр 1-го знака (s2) + 1 (Array_s2[2]): число знаков	BIN, 16 бит	массив [1..2] данных типа ANY16

## Принцип действия

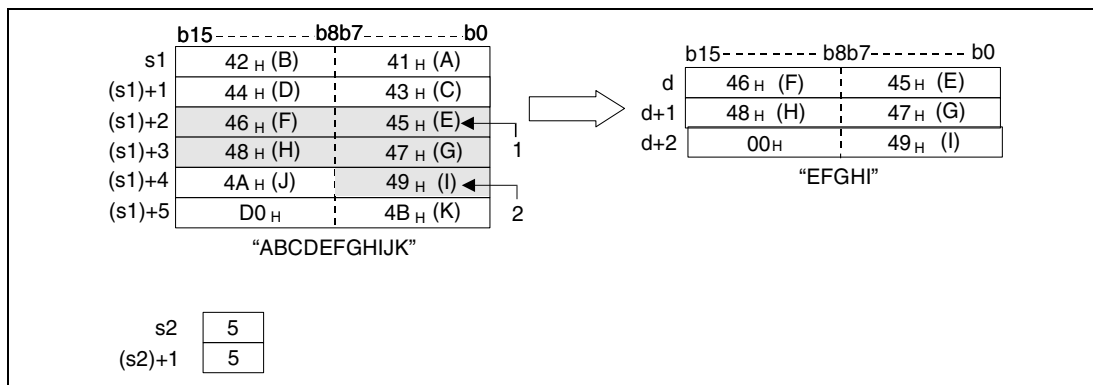
## Сохранение и перемещение фрагментов строковой величины

## MIDR Сохранение определенных фрагментов строковой величины

Команда MIDR сохраняет определенную часть строковой величины, хранящейся, начиная с  $s1$ , по адресу, начиная с  $d$ .

Первый знак сохраняемого фрагмента указан в  $s2$  ( $Array\_s2[1]$ ) как номер байта, отсчитываемый с левой стороны строковой величины (от младшего байта в  $s1$ ).

Длина сохраняемого фрагмента указана в  $s2 + 1$  ( $Array\_s2[2]$ ).

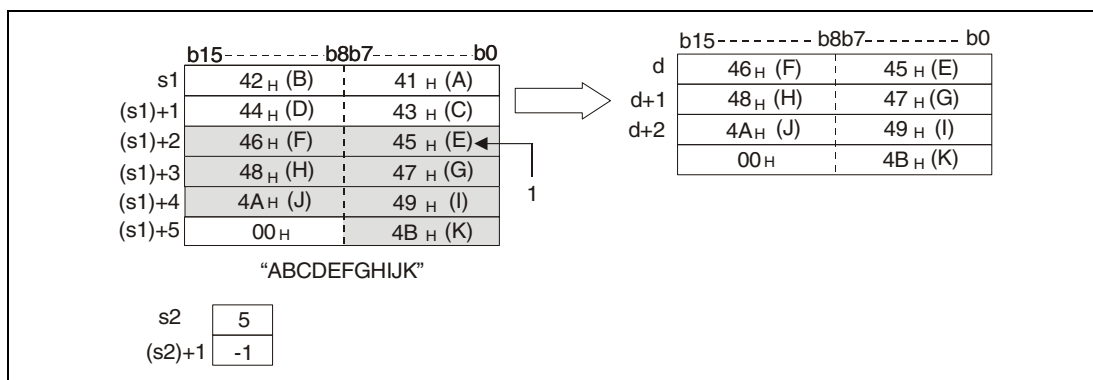


<sup>1</sup> 5-й знак ( $s2$ )

<sup>2</sup> последний сохраняемый знак

Если число знаков в  $(s2) + 1$  ( $Array\_s2[2]$ ) равно нулю, обработка не выполняется.

Если в  $(s2) + 1$  ( $Array\_s2[2]$ ) указано значение  $-1$ , то сохраняются все знаки, начиная со знака, указанного в  $s2$  ( $Array\_s2[1]$ ).



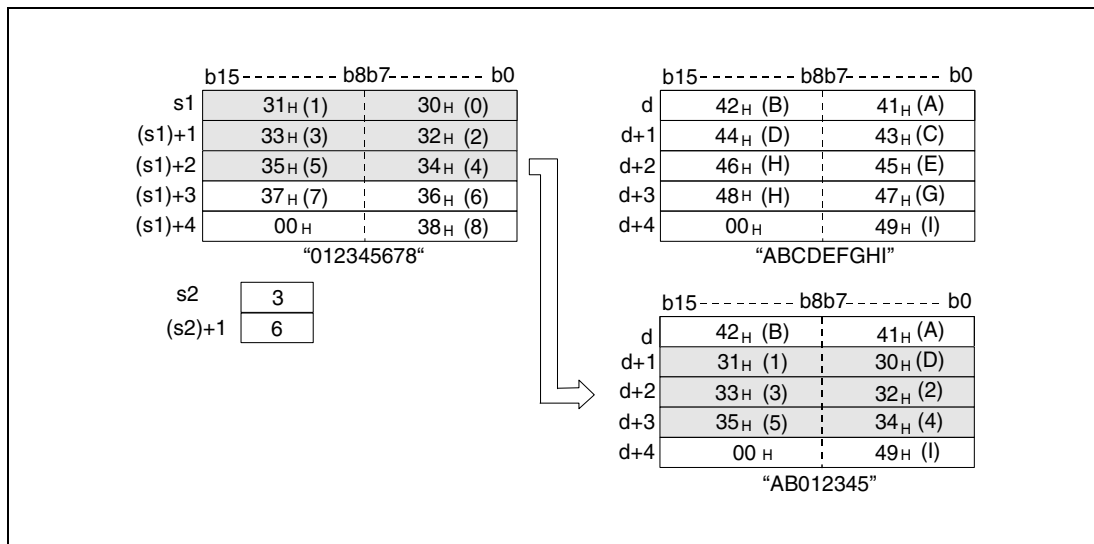
<sup>1</sup> 5-й знак ( $s2$ )

**MIDW**                    **Перемещение фрагментов строковой величины в определенную область**

Команда MIDW сохраняет фрагмент определенной длины из строковой величины, хранящейся в s1, в определенной области от d до d + n.

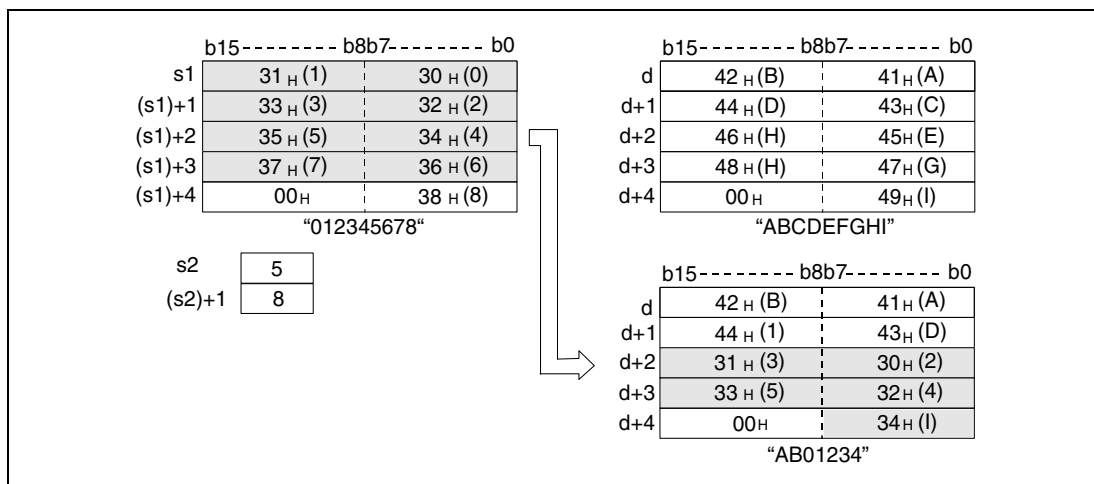
Первый адрес в области памяти от d до d + n указан в s2 (Array\_s2[1]) и отсчитывается с левой стороны строковой величины (от младшего байта в d).

Длина сохраняемого фрагмента указана в s2 + 1 (Array\_s2[2]).

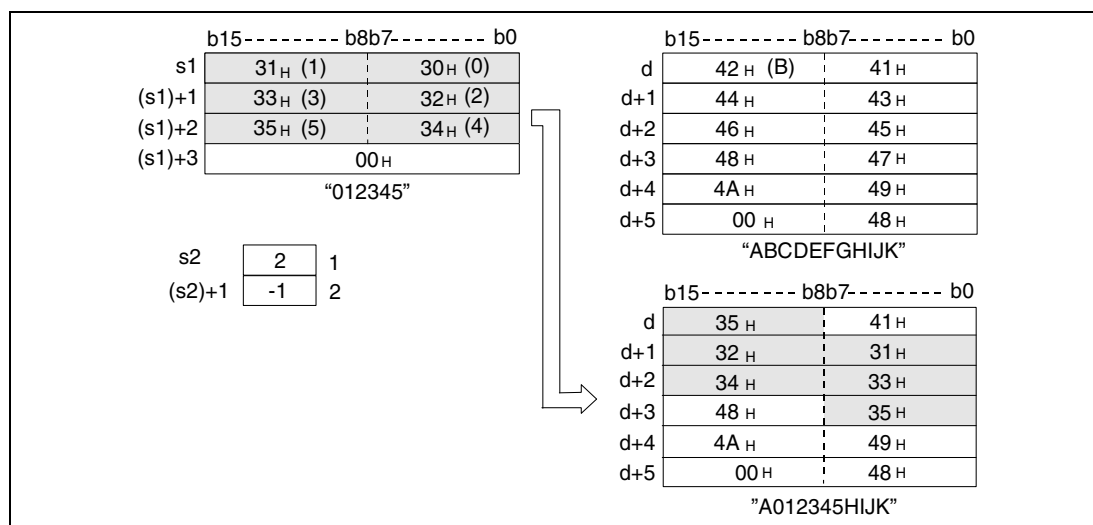


Если число знаков в (s2) + 1 (Array\_s2[2]) равно нулю, обработка не выполняется.

Если число знаков, указанное в (s2) + 1 (Array\_s2[2]), находится вне указанной области памяти, начиная с d, остаток строковой величины отбрасывается. В следующем примере (см. иллюстрацию) знаки от "35H" до "37H" не сохраняются.



Если в  $(s2) + 1$  (`Array_s2[2]`) указано значение  $-1$ , то сохраняются все знаки строковой величины, начиная с  $s1$ .



#### Источники ошибок

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

Команда MIDR

- Указанный в  $s2$  (`Array_s2[1]`) начальный адрес сохраняемых знаков находится вне области от  $s1$  до  $(s1) + n$  (код ошибки 4101).
- Указанное в  $(s2) + 1$  (`Array_s2[2]`) количество сохраняемых знаков находится вне области памяти от  $d$  до  $d + n$  (код ошибки 4101).

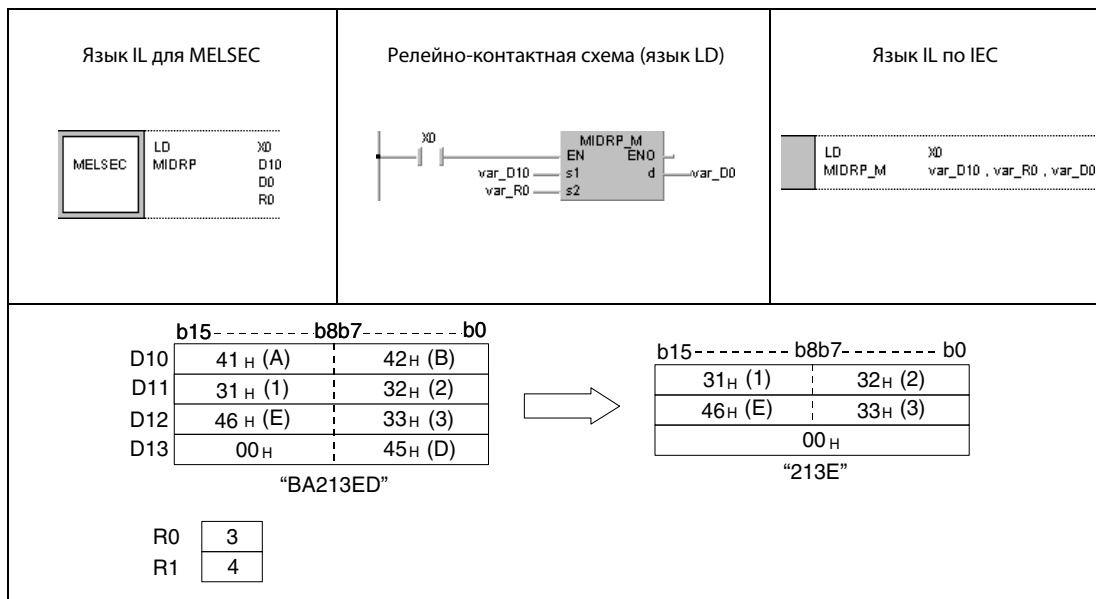
Команда MIDW

- Указанный в  $s2$  (`Array_s2[1]`) начальный адрес области памяти находится вне области от  $d$  до  $d + n$  (код ошибки 4101).
- Указанное в  $(s2) + 1$  (`Array_s2[2]`) количество сохраняемых знаков находится вне области памяти от  $s1$  до  $(s1) + n$  (код ошибки 4101).



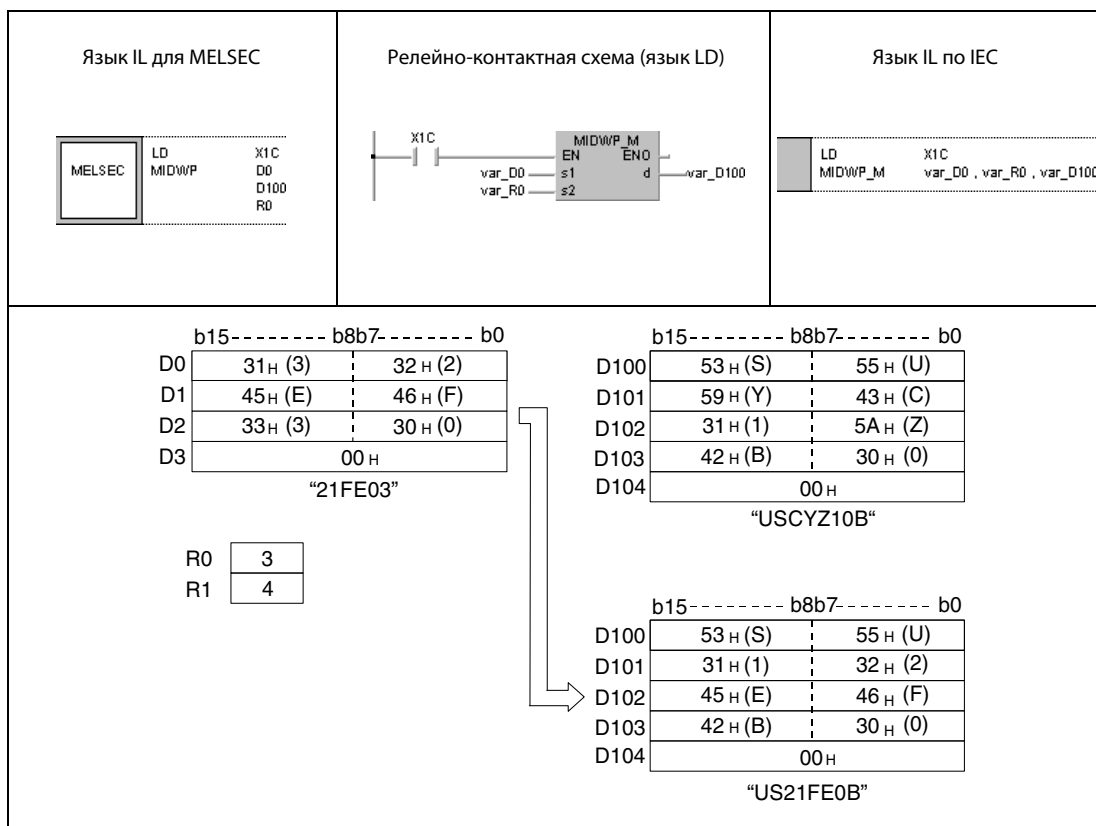
**Пример 1** MIDRP

Следующая программа при положительном фронте сигнала X0 сохраняет в регистрах с D0 по D2 указанное в R1 (var\_R0 Array [2]) количество знаков строковой величины, хранящейся в регистрах с D10 по D13, начиная с позиции, указанной в R0 (var\_R0 Array [1]).



**Пример 2** MIDWP

Следующая программа при положительном фронте сигнала X1C сохраняет знаки строковой величины, хранящейся с D0 по D3, в регистрах с D100 по D104. Количество этих знаков указано в R1 (var\_R0 Array [2]), а номер первого считываемого знака – в R0 (var\_R0 Array [1]).



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке ПОУ эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.11.18 INSTR, INSTRP

**Процессор**

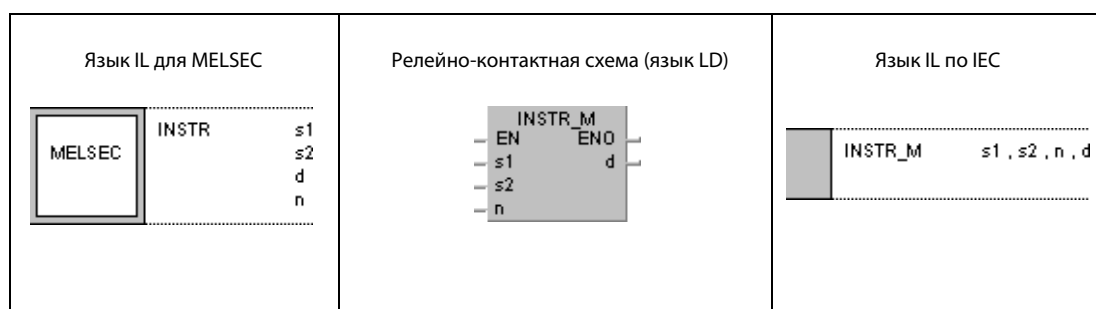
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

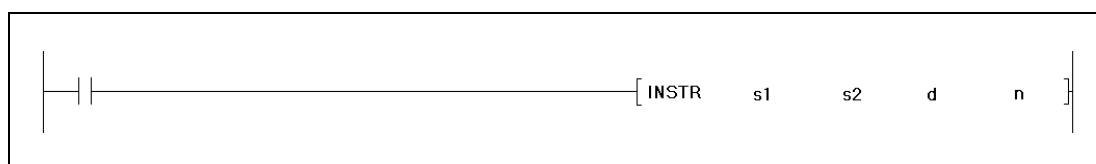
**Операнды MELSEC Q**

	Операнды										Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непер. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы		Иные		
	битовые	словные		битовые	словные			K, H (16#)	\$			
s1	—	●	●	—	—	—	—	—	●	—	SM0	5
s2	—	●	●	—	—	—	—	—	●	—		
d	●	●	●	●	●	●	●	—	—	—		
n	●	●	●	●	●	●	●	●	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s1	Первый адрес операндов, в котором сохранена искомая строковая величина.	строковая величина
s2	Первый адрес операндов строковой величины, в которой ведется поиск.	
d	Первый адрес, в котором сохраняется результат поиска.	BIN, 16 бит
n	Начальный адрес поиска	

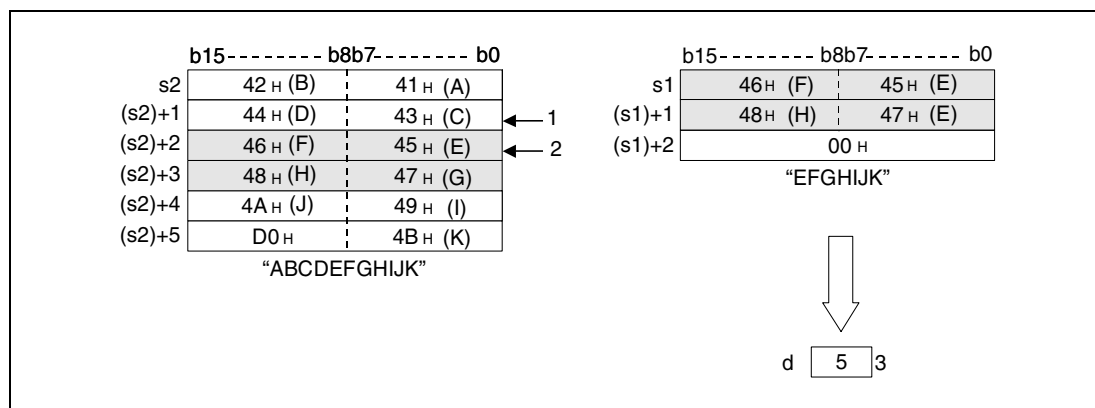
**Принцип действия****Поиск строковых величин****INSTR Поиск строковой величины**

Команда INSTR ищет в строковой величине, хранящейся в операндах с  $s2$  по  $(s2) + n$ , строковую величину, указанную в операндах с  $s1$  по  $(s1) + n$ .

Поиск начинается со знака, указанного в  $n$ .

Номер знака, начиная с которого обнаружена искомая строковая величина, сохраняется в  $d$ . Номер знака указывается по отношению к левой части строковой величины (младшему байту  $s2$ ).

В случае  $n = 3$



<sup>1</sup> Поиск начинается с 3-го знака.

<sup>2</sup> первый знак искомой строковой величины

<sup>3</sup> результат поиска

Если строковая величина не найдена, в  $d$  сохраняется ноль.

Если в  $n$  указано отрицательное значение или ноль, команда не выполняется.

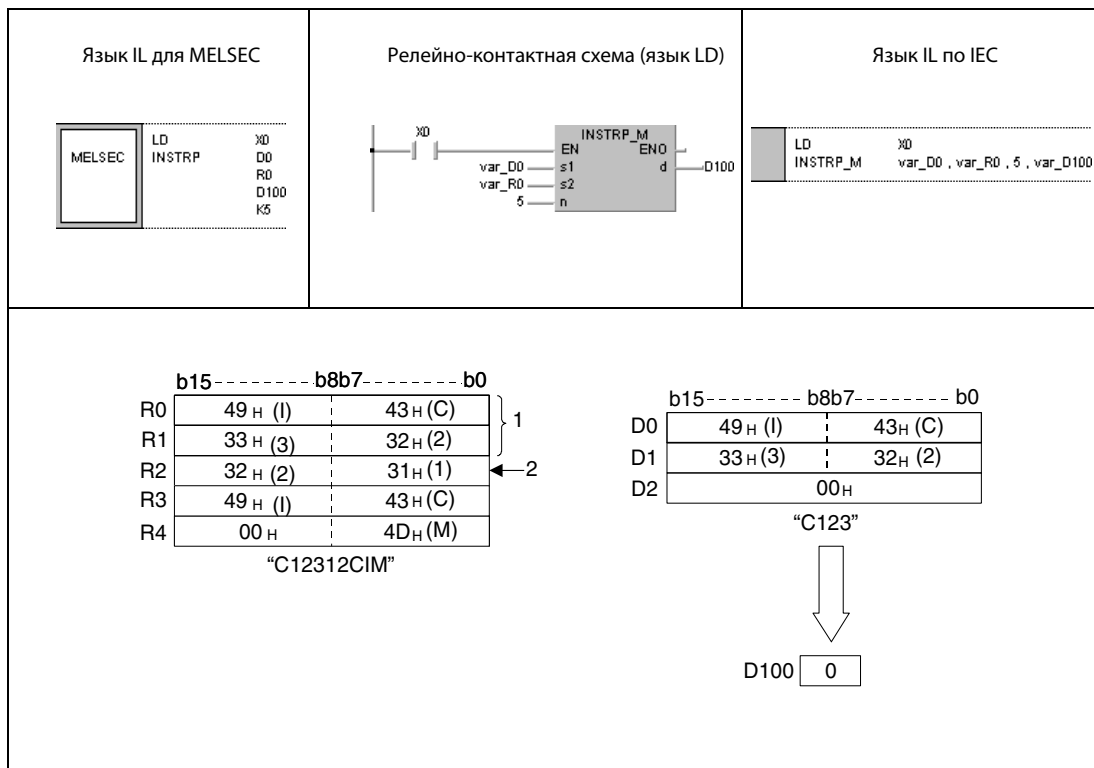
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанный в  $n$  начальный адрес поиска находится вне области от  $s2$  до  $(s2) + n$  (код ошибки 4100).

**Пример 1** INSTRP

Следующая программа при положительном фронте сигнала X0 ищет в области, начиная с R0, строковую величину, указанную в регистрах с D0 по D2. Поиск начинается с 5-го знака строковой величины, хранящейся, начиная с R0. Результат поиска (0) сохраняется в D100.

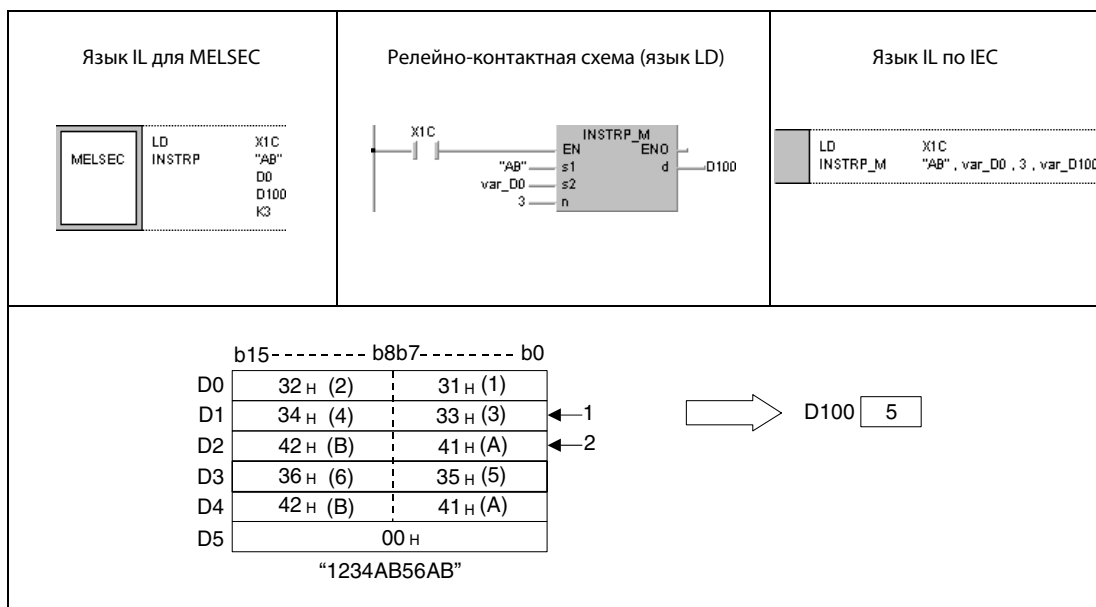


<sup>1</sup> В этой области поиск не ведется, так как поиск начинается лишь с 5-го знака.

<sup>2</sup> Поиск начинается с 5-го знака.

**Пример 2** INSTRP

Следующая программа при положительном фронте сигнала X0 ищет в данных, хранящихся, начиная с D0, строковую величину "AB". Поиск начинается с 3-го знака строковой величины, хранящейся, начиная с D0. Результат поиска (5) сохраняется в D100.



<sup>1</sup> начало поиска (3-й знак)

<sup>2</sup> Искомая строковая величина начинается с 5-го знака.

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.11.19 EMOD, EMODP

**Процессор**

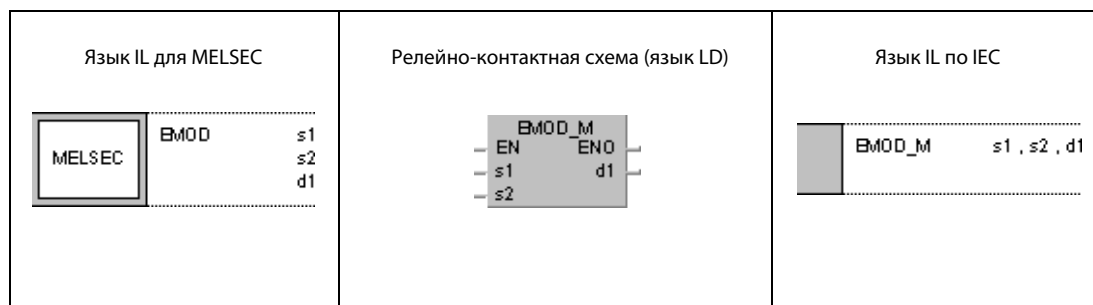
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

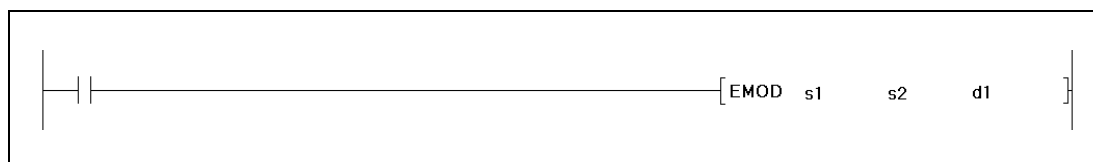
**Операнды MELSEC Q**

	Операнды										Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы		Иные		
	битовые	словные		битовые	словные			K, H (16#)	E			
s1	—	●	●	—	●	●	—	—	●	—	SMO	4
s2	●	●	●	●	●	●	●	●	—	—		
d1	—	●	●	—	—	—	—	—	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

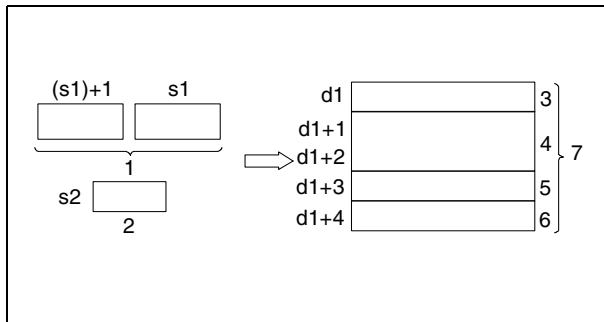
Операнд	Значение	Тип данных
s1	Число с плавающей запятой (данные типа REAL) или первый адрес операнда, в котором сохранено число с плавающей запятой.	REAL
s2	Количество разрядов, на которое десятичная запятая смещается вправо, или первый адрес операнда, в котором сохранены эти данные.	BIN, 16 бит
d1	Первый адрес операнда, в котором сохраняется число с плавающей запятой в формате BCD.	

**Принцип действия**

**Преобразование числа с плавающей запятой в двоично-десятичный формат**

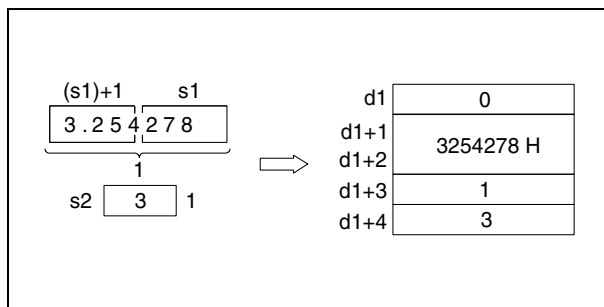
**EMOD Преобразование в двоично-десятичный формат**

Команда EMOD преобразует в двоично-десятичный формат число с плавающей запятой (данные типа REAL), указанное в s1 и (s1) + 1, с учетом указанного в s2 смещения запятой вправо. Результат сохраняется в операндах с d1 по (d1) + 4.

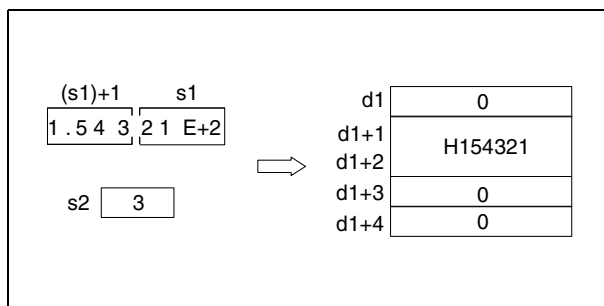
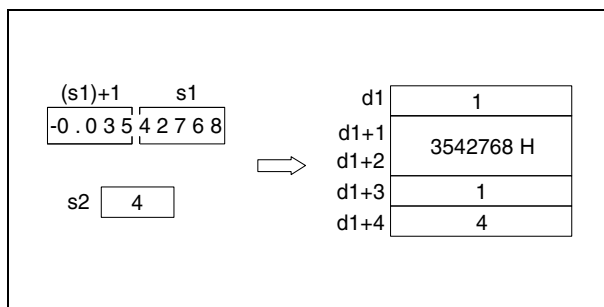


- 1 число с плав. запятой (данные типа REAL)
- 2 перемещение десятич. запятой вправо
- 3 арифметический знак (0 = положительный/1 = отрицательный)
- 4 7 двоично-десятичных знаков
- 5 арифметический знак показателя степени (0 = положительный/1 = отрицательный)
- 6 показатель степени в двоично-десятичном виде (диапазон от 0 до 38)
- 7 число с плав. запятой в формате BCD

Ниже изображены примеры преобразований.

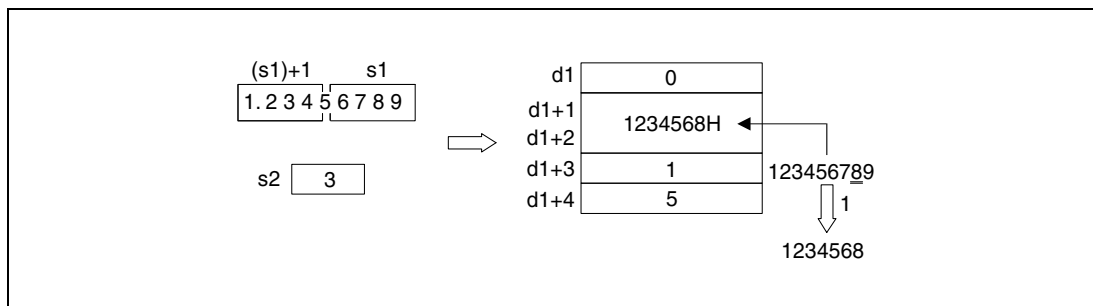


- 1 число с плав. запятой (данные типа REAL)





Число с плавающей запятой в  $s1$  и  $(s1) + 1$  округляется до 7 знаков, а затем сохраняется в  $(d1) + 1$  и  $(d1) + 2$ .



<sup>1</sup> Округляется вверх.

**Источники ошибок**

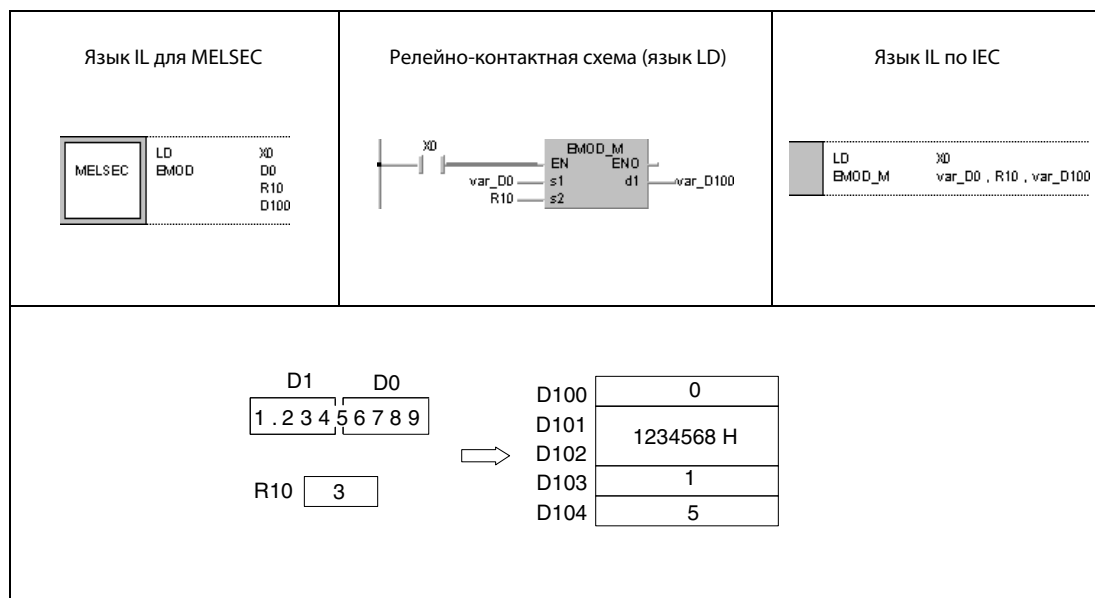
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Количество разрядов, на которое смещается запятая  $s2$ , находится вне диапазона от 0 до 7 (код ошибки 4100).
- Значение, указанное в операндах от  $d1$  до  $(d1) + 4$ , находится вне области памяти операнда (код ошибки 4101).

**Пример**

## EMOD

Следующая программа при включенном входе X0 преобразует число с плавающей запятой (данные типа REAL), указанное в D0 и D1, с учетом указанного в R10 смещения запятой и сохраняет результат в регистрах с D100 по D104.

**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.11.20 EREXP, EREXPP

**Процессор**

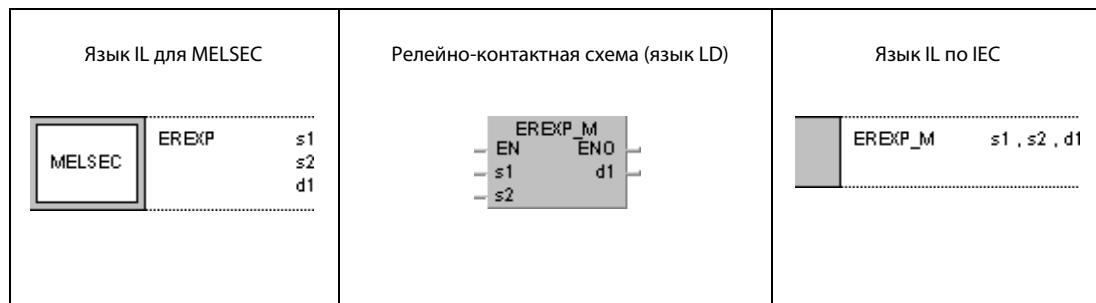
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

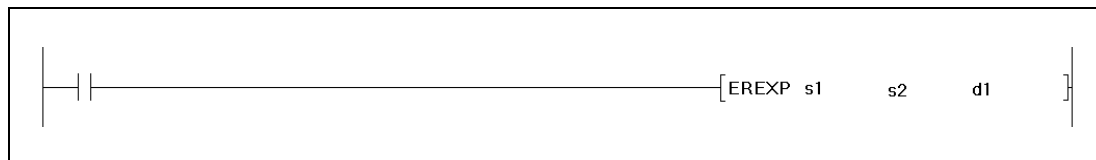
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□G□□	Индексные регистры Zn	Константы K, H (16#)			Иные U
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	SM0	4	
s2	●	●	●	●	●	●	●	—			
d1	—	●	●	—	●	●	—	—			

**GX IEC Developer**



**GX Developer**



**Переменные**

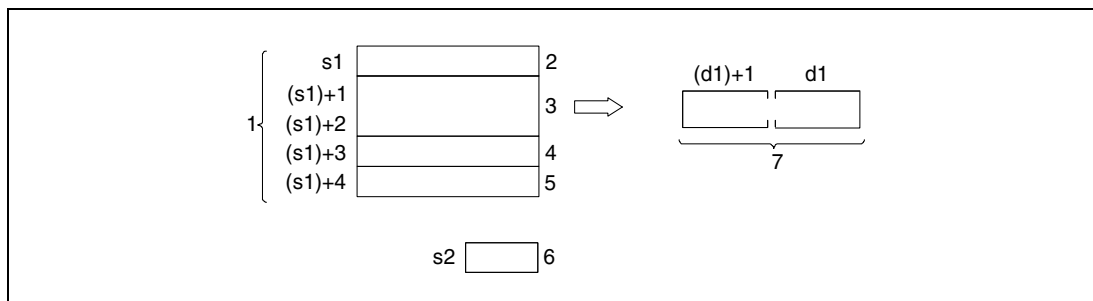
Операнд	Значение	Тип данных
s1	Первый адрес операнда, в котором сохранено число с плавающей запятой в формате BCD.	BIN, 16 бит
s2	Данные дробной части или операнд, в котором сохранены эти данные.	
d1	Операнд, в котором сохраняется число с плавающей запятой (данные типа REAL).	REAL

**Принцип действия**

**Преобразование числа с плавающей запятой в десятичный формат**

**EREXP Преобразование в десятичный формат**

Команда EREXP преобразует число с плавающей запятой в двоично-десятичном формате (BCD), хранящееся с операндах с s1 по (s1) + 4, в число с плавающей запятой (данные типа REAL) в десятичном формате с учетом указанного в s2 количества разрядов дробной части. Результат сохраняется в d1 и (d1) + 1.

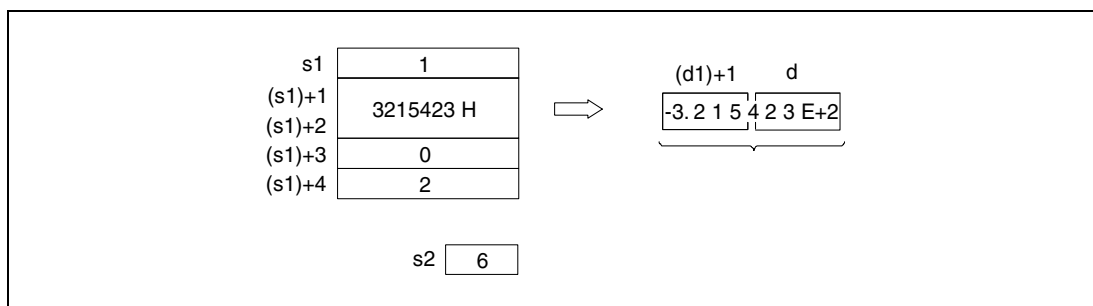


- <sup>1</sup> число с плавающей запятой в формате BCD
- <sup>2</sup> арифметический знак (0 = положительный/1 = отрицательный)
- <sup>3</sup> 7 двоично-десятичных знаков
- <sup>4</sup> арифметический знак показателя степени (0 = положительный/1 = отрицательный)
- <sup>5</sup> показатель степени в двоично-десятичном виде (диапазон от 0 до 38)
- <sup>6</sup> количество разрядов дробной части (диапазон от 0 до 7)
- <sup>7</sup> число с плавающей запятой (данные типа REAL)

В качестве положительного арифметического знака числа в s1 и арифметического знака показателя степени в (s1) + 3 устанавливаются 0. В качестве отрицательного арифметического знака устанавливается 1.

Показатель степени в формате BCD (s1) + 4 может находиться в диапазоне между 0 и 38.

Дробная часть может иметь от 0 до 7 разрядов (s2).



**Источники ошибок**

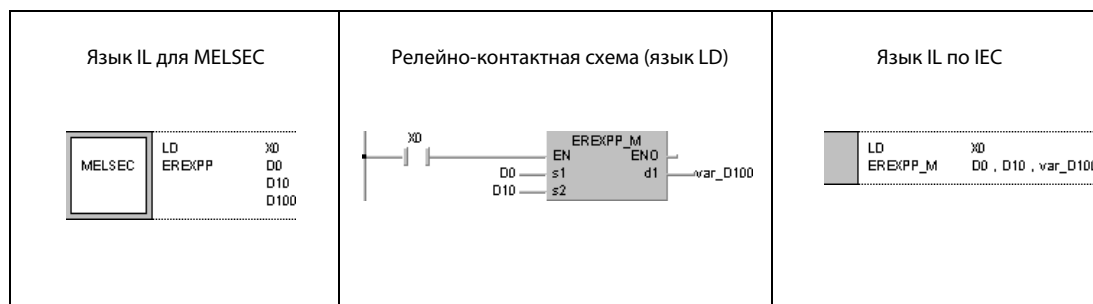
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В качестве арифметического знака в s1 указано иное значение кроме 0 или 1 (код ошибки 4100).
- Двоично-десятичные данные в (s1) + 1 и (s1) + 2 имеют более 8 разрядов (код ошибки 4100).
- Арифметический знак показателя степени в (s1) + 3 имеет иное значение кроме 0 или 1 (код ошибки 4100).
- Данные показателя степени в (s1) + 4 находятся вне диапазона от 0 до 38 (код ошибки 4100).
- Количество разрядов дробной части в s2 находится вне диапазона от 0 до 7 (код ошибки 4101).

**Пример**

EREXPP

Следующая программа при положительном фронте сигнала X0 преобразует число с плавающей запятой в двоично-десятичном формате, хранящееся в операндах с D0 по D4, в число с плавающей запятой (данные типа REAL) в десятичном формате с учетом указанного в D10 количества разрядов дробной части и сохраняет результат по адресу D100 и D101.



## 7.12 Особые функции

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе IEC
Вычисление синуса	SIN	SIN_MD
		SIN_E_MD
	SINP	SIN_P_MD
		SIN_P_E_MD
Вычисление косинуса	COS	COS_MD
		COS_E_MD
	COSP	COS_P_MD
		COS_P_E_MD
Вычисление тангенса	TAN	TAN_MD
		TAN_E_MD
	TANP	TAN_P_MD
		TAN_P_E_MD
Вычисление арксинуса	ASIN	ASIN_MD
		ASIN_E_MD
	ASINP	ASIN_P_MD
		ASIN_P_E_MD
Вычисление арккосинуса	ACOS	ACOS_MD
		ACOS_E_MD
	ACOSP	ACOS_P_MD
		ACOS_P_E_MD
Вычисление арктангенса	ATAN	ATAN_MD
		ATAN_E_MD
	ATANP	ATAN_P_MD
		ATAN_P_E_MD
Пересчет из градусов в радианы	RAD	RAD_MD
		RAD_E_MD
	RADP	RAD_P_MD
		RAD_P_E_MD
Пересчет из радиан в градусы	DEG	DEG_MD
		DEG_E_MD
	DEGP	DEG_P_MD
		DEG_P_E_MD
Вычисление квадратного корня	SQR	SQR_MD
		SQR_E_MD
	SQRP	SQR_P_MD
		SQR_P_E_MD
Вычисление экспоненты	EXP	EXP_MD
		EXP_E_MD
	EXPP	EXP_P_MD
		EXP_P_E_MD

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе IEC
Вычисление натурального логарифма	LOG	LOG_MD
		LOG_E_MD
	LOGP	LOG_P_MD
		LOG_P_E_MD
Генерирование случайных чисел	RND	RND_M
	RNDP	RNDP_M
Обновление серий случайных чисел	SRND	SRND_M
	SRNDP	SRNDP_M
Вычисление квадратного корня из 4-значных двоично-десятичных данных	BSQR	BSQR_MD
		BSQR_K_MD
	BSQRP	BSQR_P_MD
		BSQR_K_P_MD
Вычисление квадратного корня из 8-значных двоично-десятичных данных	BDSQR	BDSQR_MD
		BDSQR_K_MD
	BDSQRP	BDSQR_P_MD
		BDSQR_K_P_MD
Вычисление синуса двоично-десятичных данных	BSIN	BSIN_MD
		BSIN_K_MD
	BSINP	BSIN_P_MD
		BSIN_K_P_MD
Вычисление косинуса двоично-десятичных данных	BCOS	BCOS_MD
		BCOS_K_MD
	BCOSP	BCOS_P_MD
		BCOS_K_P_MD
Вычисление тангенса двоично-десятичных данных	BTAN	BTAN_MD
		BTAN_K_MD
	BTANP	BTAN_P_MD
		BTAN_K_P_MD
Вычисление арксинуса двоично-десятичных данных	BASIN	BASIN_MD
	BASINP	BASIN_P_MD
Вычисление арккосинуса двоично-десятичных данных	BACOS	BACOS_MD
	BACOSP	BACOS_P_MD
Вычисление арктангенса двоично-десятичных данных	BATAN	BATAN_MD
	BATANP	BATAN_P_MD

**ПРИМЕЧАНИЕ** В редакторах по стандарту IEC используйте стандартные команды IEC.

### 7.12.1 SIN, SINP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

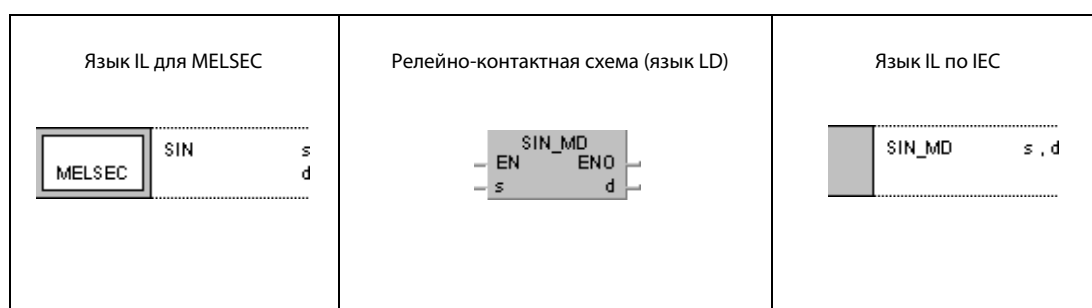
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

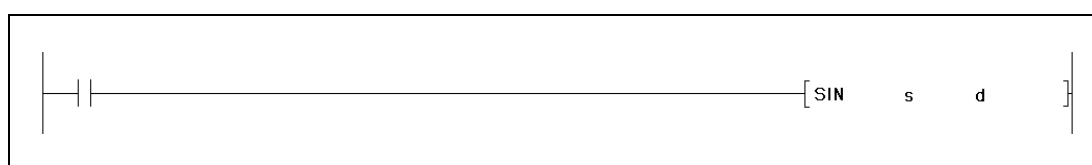
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непос. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы E			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	●	●	—	●	—	3	
d	—	●	●	—	●	●	—	—	—		

**GX IEC Developer**



**GX Developer**



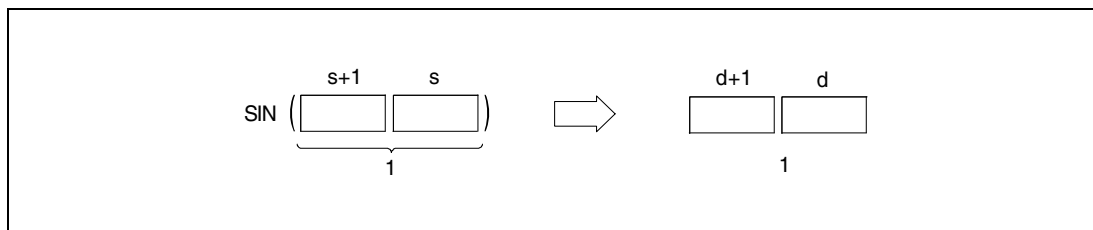
**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранены данные угла, для которого вычисляется синус.	REAL
d	Первый адрес операнда, в котором сохраняется результат.	



**Принцип действия****Вычисление синуса****SIN                      Вычисление синуса чисел с плавающей запятой**

Команда SIN вычисляет синус угла, указанного в  $s$  и  $s + 1$ , и сохраняет результат в  $d$  и  $d + 1$ .



<sup>1</sup> число с плавающей запятой (данные типа REAL)

Угол в  $s$  и  $s + 1$  должен быть указан в радианах (градусов  $\times \pi/180$ ). Преобразование между градусами и радианами более подробно разъяснено в разделах, посвященных командам RAD и DEG.

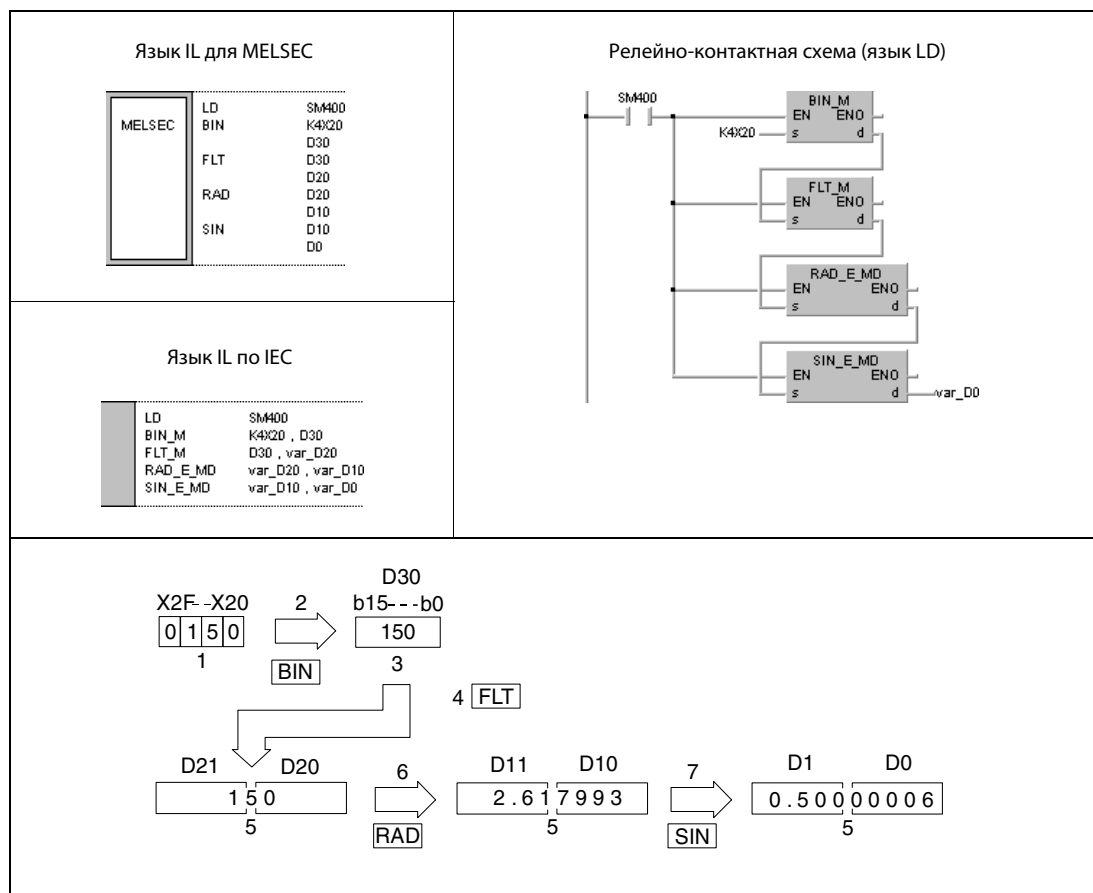
**Источники ошибок**

Если применяется процессор Q4AR и при этом указанный в  $s$  адрес содержит  $-0$  и специальный маркер SM707 не установлен, выводится код ошибки 4100.

## Пример

## SIN

Следующая программа при включенном маркере SM400 вычисляет синус 4-значного значения угла в двоично-десятичном виде, указанного в операндах с X20 по X2F, и сохраняет результат в виде числа с плавающей запятой (данных типа REAL) в D0 и D1.



- 1 двоично-десятичное значение
- 2 преобразование в формат BIN
- 3 двоичное значение
- 4 преобразование в формат с плавающей запятой
- 5 число с плавающей запятой (данные типа REAL)
- 6 преобразование в радианы
- 7 вычисление синуса

## ПРИМЕЧАНИЕ

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.12.2 COS, COSP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

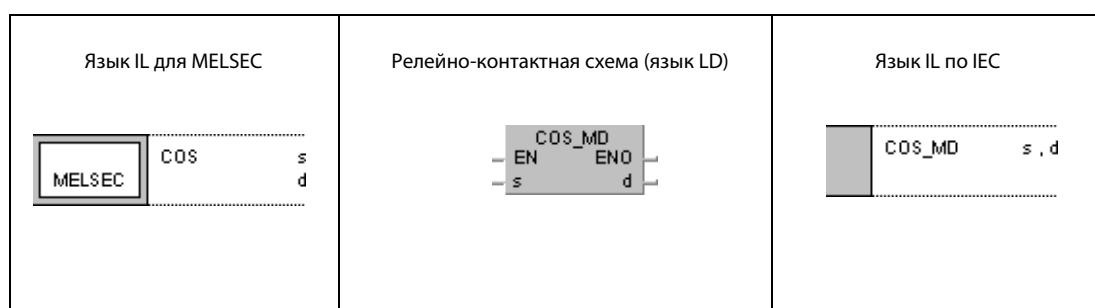
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

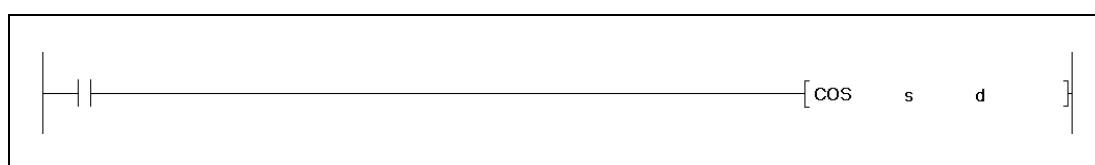
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы E			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	●	●	—	●	—	3	
d	—	●	●	—	●	●	—	—	—		

**GX IEC Developer**



**GX Developer**

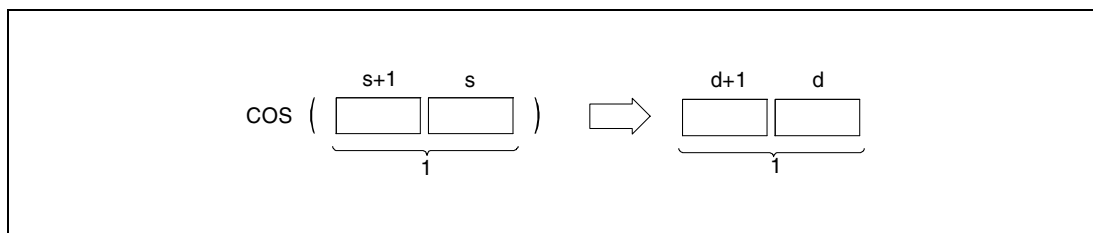


**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда с данными угла, для которого вычисляется косинус.	REAL
d	Первый адрес операнда, в котором сохраняется результат.	

**Принцип действия****Вычисление косинуса****COS****Вычисление косинуса числа с плавающей запятой**

Команда COS вычисляет косинус угла, указанного в s и s + 1, и сохраняет результат в d и d + 1.



<sup>1</sup> число с плавающей запятой (данные типа REAL)

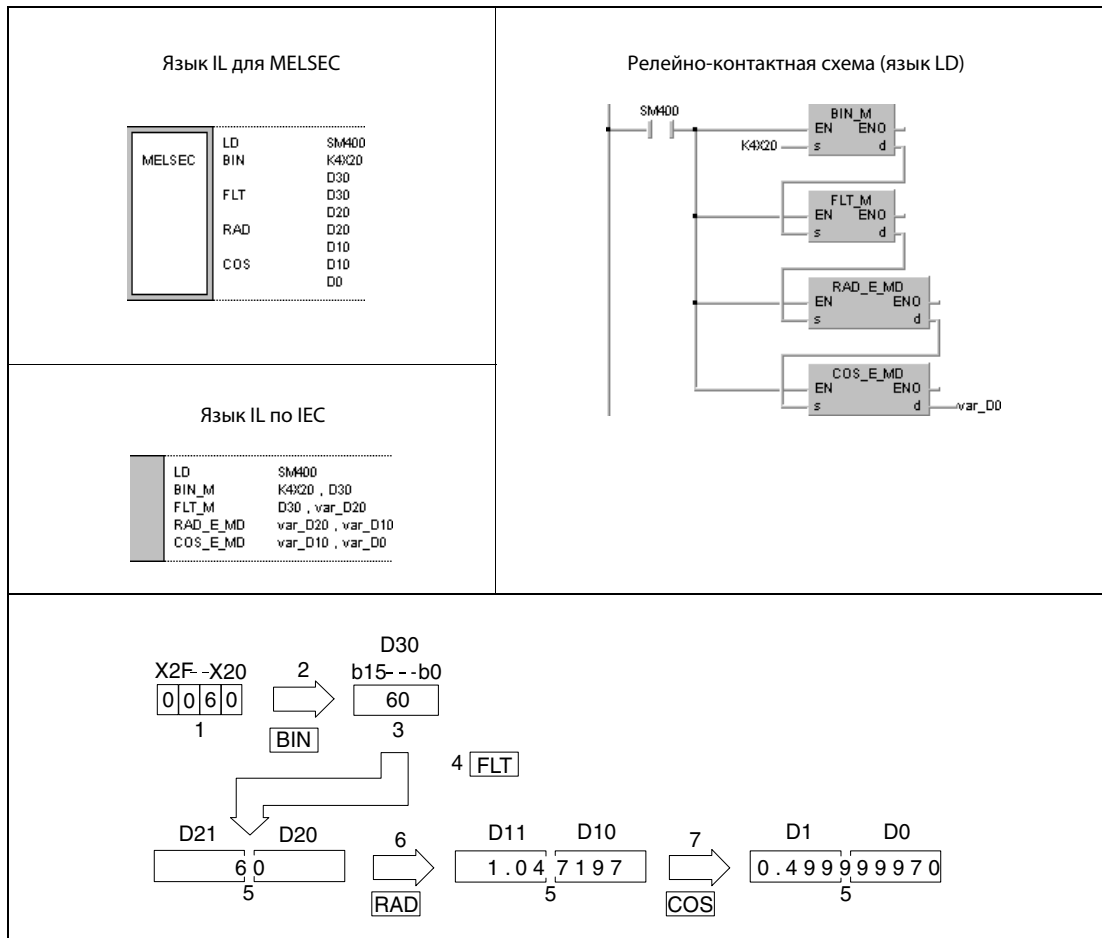
Угол в s и s + 1 должен быть указан в радианах (градусов  $\times \pi/180$ ). Преобразование между градусами и радианами более подробно разъяснено в разделах, посвященных командам RAD и DEG.

**Источники ошибок**

Если применяется процессор Q4AR и при этом указанный в s адрес содержит -0 и специальный маркер SM707 не установлен, выводится код ошибки 4100.

**Пример** COS

Следующая программа при включенном маркере SM400 вычисляет косинус угла, указанного в операндах с X20 по X2F в 4-значном двоично-десятичном виде, и сохраняет результат в виде числа с плавающей запятой (данных типа REAL) в D0 и D1.



- 1 двоично-десятичное значение
- 2 преобразование в двоичный формат
- 3 двоичное значение
- 4 преобразование в формат с плавающей запятой
- 5 число с плавающей запятой (данные типа REAL)
- 6 пересчет в радианы
- 7 вычисление косинуса

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.12.3 TAN, TANP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

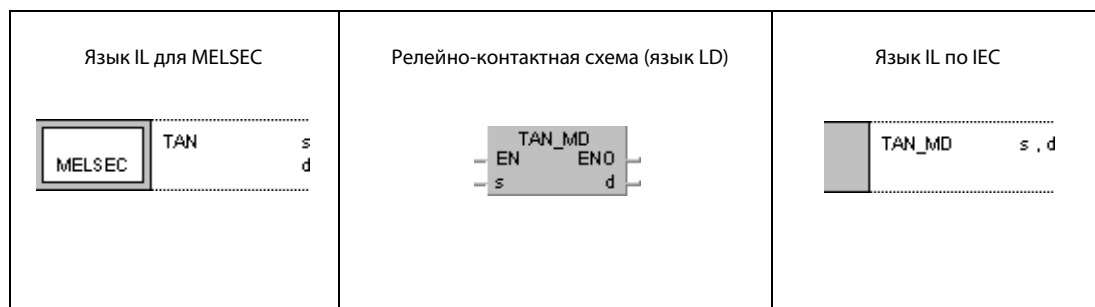
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

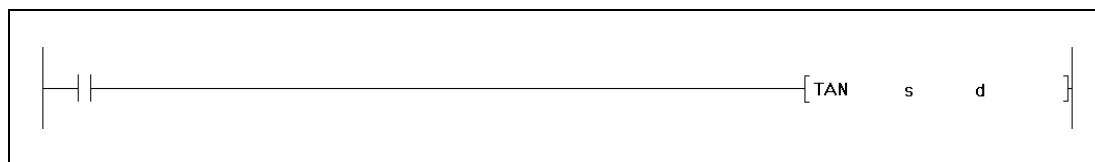
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□□	Индексные регистры Zn	Константы E			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	●	●	—	●	—	SM0	3
d	—	●	●	—	●	●	—	—	—		

**GX IEC Developer**



**GX Developer**

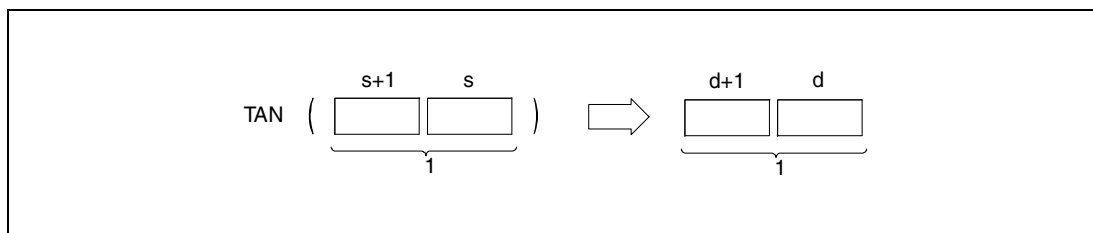


**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда с данными угла, для которого вычисляется тангенс.	REAL
d	Первый адрес операнда, в котором сохраняется результат.	

**Принцип действия****Вычисление тангенса****TAN Вычисление тангенса числа с плавающей запятой**

Команда TAN вычисляет тангенс угла, указанного в  $s$  и  $s + 1$ , и сохраняет результат в  $d$  и  $d + 1$ .



<sup>1</sup> число с плавающей запятой (данные типа REAL)

Угол в  $s$  и  $s + 1$  должен быть указан в радианах (градусов  $\times \pi/180$ ). Преобразование между градусами и радианами более подробно разъяснено в разделах, посвященных командам RAD и DEG.

Если в  $s$  и  $s + 1$  указан угол, равный  $\pi/2$  радиан или  $(3/2) \times \pi$  радиан, вырабатывается сообщение об ошибке.

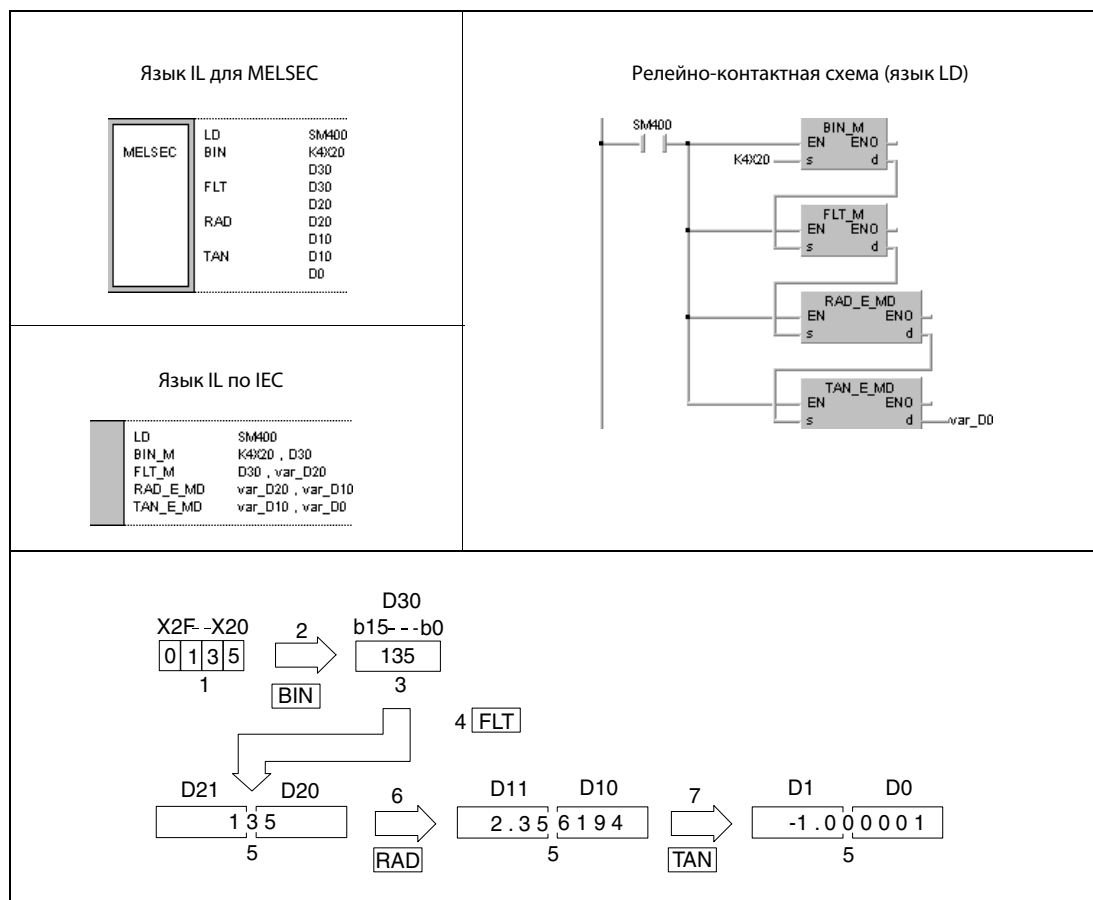
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Результат обработки равен нулю или находится вне диапазона от  $\pm 2^{-127}$  до  $\pm 2^{129}$  (код ошибки 4100).
- Если применяется процессор Q4AR и при этом указанный в  $s$  адрес содержит  $-0$  и специальный маркер SM707 не установлен, выводится код ошибки 4100.

## Пример TAN

Следующая программа при положительном фронте сигнала SM400 вычисляет тангенс угла, указанного в 4-значном двоично-десятичном виде в операндах с X20 по X2F, и сохраняет результат в виде числа с плавающей запятой (данных типа REAL) в D0 и D1.



- 1 двоично-десятичное значение
- 2 преобразование в двоичный формат
- 3 двоичное значение
- 4 преобразование в формат с плавающей запятой
- 5 число с плавающей запятой (данные типа REAL)
- 6 пересчет в радианы
- 7 вычисление тангенса

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.



### 7.12.4 ASIN, ASINP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

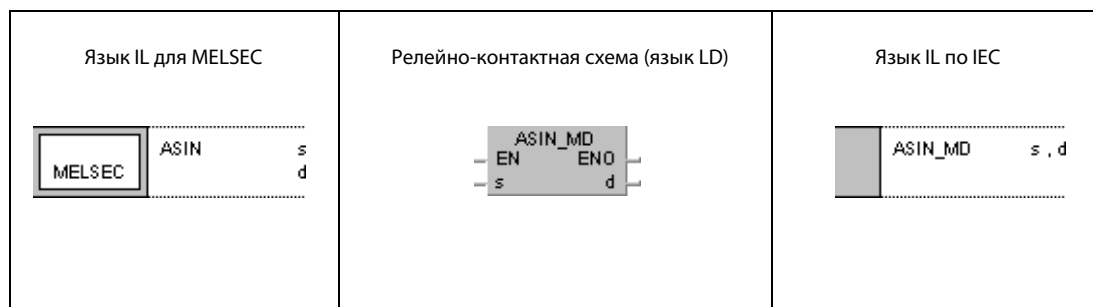
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

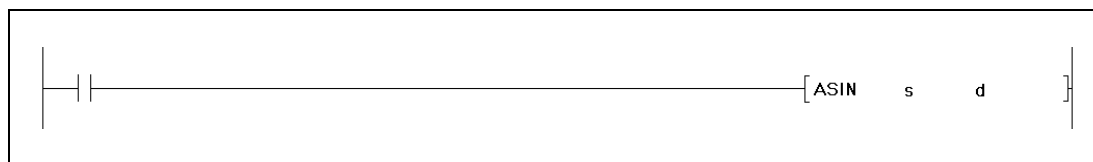
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□□	Индексные регистры Zn	Константы E			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	●	●	—	●	—	SM0	3
d	—	●	●	—	●	●	—	—	—		

**GX IEC Developer**



**GX Developer**

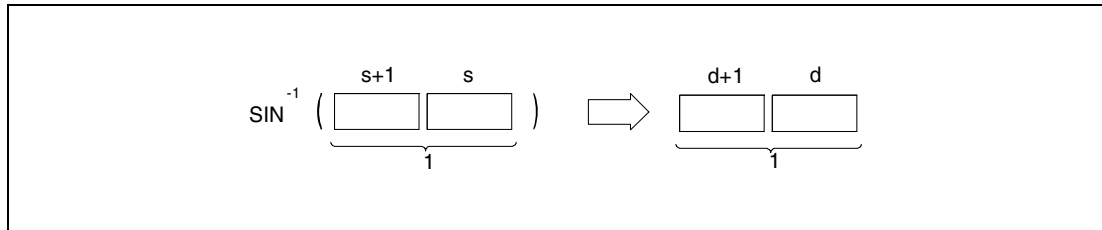


**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда, содержащего синус, для которого вычисляется арксинус.	REAL
d	Первый адрес операнда, в котором сохраняется результат.	

**Принцип действия****Вычисление арксинуса****ASIN****Вычисление арксинуса числа с плавающей запятой**

Команда ASIN вычисляет угол по указанному s и s+1 синусу и сохраняет результат в d и d+1.



<sup>1</sup> число с плавающей запятой (данные типа REAL)

Указанный в s и s + 1 синус может находиться в диапазоне от –1 до 1.

Угол в d и d + 1 выражен в радианах (градусов  $\times \pi/180$ ). Преобразование между градусами и радианами более подробно разъяснено в разделах, посвященных командам RAD и DEG.

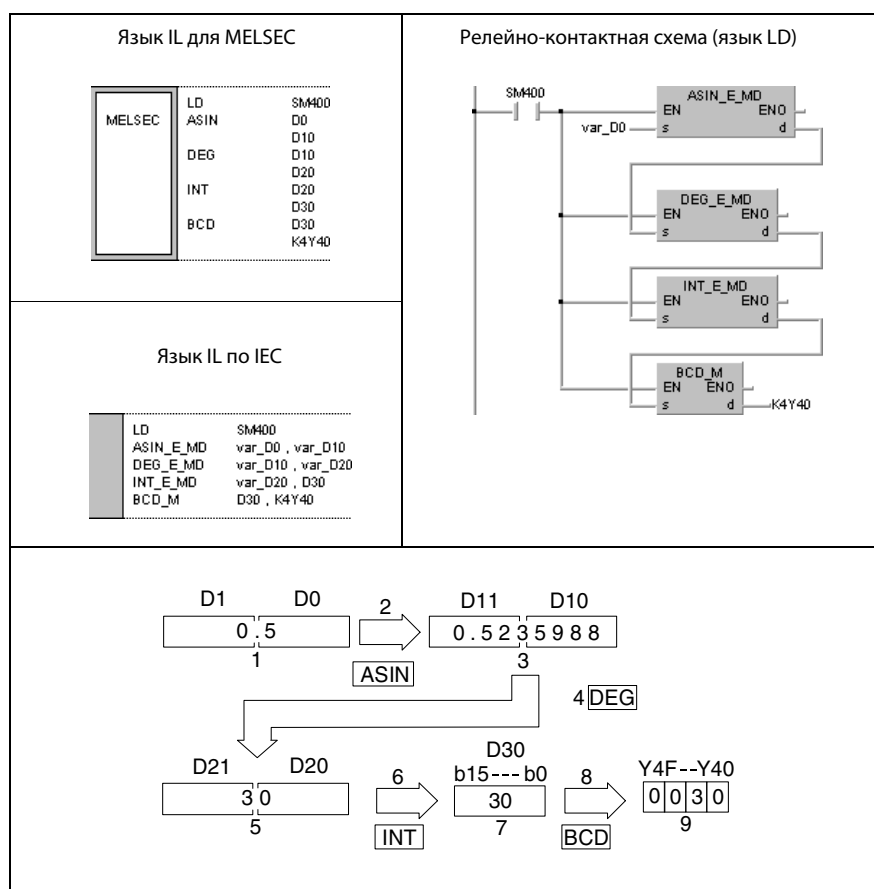
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Значение в s и s + 1 находится вне диапазона от –1 до 1 (код ошибки 4100).
- Если применяется процессор Q4AR и при этом указанный в s адрес содержит –0 и специальный маркер SM707 не установлен, выводится код ошибки 4100.

**Пример** ASIN

Следующая программа при включенном маркере SM400 вычисляет арксинус числа с плавающей запятой (данных типа REAL), хранящегося в D0 и D1, и выводит полученный угол в радианах в виде 4-значного двоично-десятичного значения на выходы с Y20 по Y4F.



- 1 число с плавающей запятой (данные типа REAL)
- 2 вычисление арксинуса
- 3 число с плавающей запятой (данные типа REAL)
- 4 преобразование угла
- 5 число с плавающей запятой (данные типа REAL)
- 6 преобразование в двоичный формат
- 7 двоичное значение
- 8 преобразование в двоично-десятичный формат
- 9 двоично-десятичное значение

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке ПОУ этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.12.5 ACOS, ACOSP

**Процессор**

AnS	AnN	AnA(S)	AnU	Qn(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

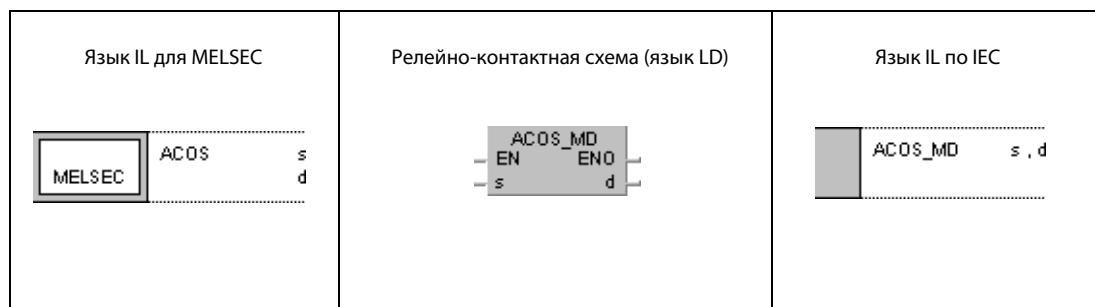
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

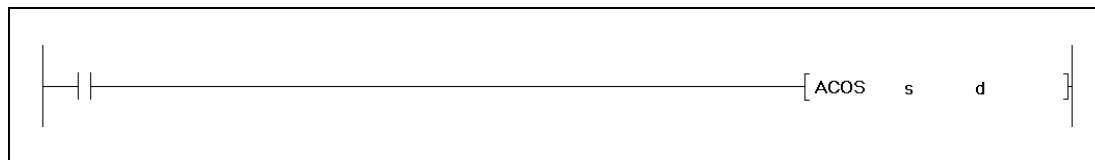
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы E			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	●	●	—	●	—	SM0	3
d	—	●	●	—	●	●	—	—	—		

**GX IEC Developer**



**GX Developer**

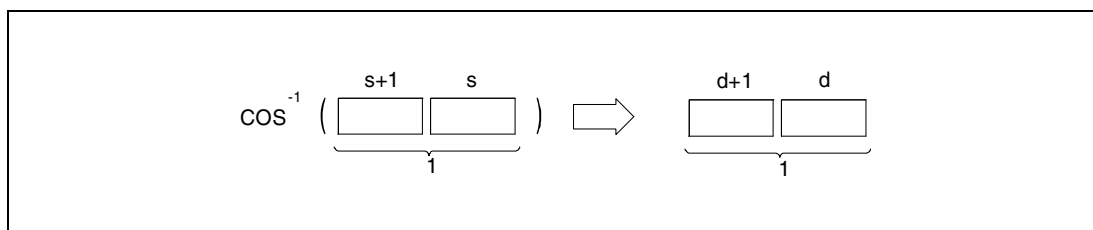


**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда, содержащего косинус, для которого вычисляется арккосинус.	REAL
d	Первый адрес операнда, в котором сохраняется результат.	

**Принцип действия****Вычисление арккосинуса****ACOS****Вычисление арккосинуса числа с плавающей запятой**

Команда ACOS вычисляет угол по указанному в s и s + 1 косинусу и сохраняет результат в d и d + 1.



<sup>1</sup> число с плавающей запятой (данные типа REAL)

Косинус в s и s + 1 может находиться в диапазоне от -1 до 1.

Угол в d и d + 1 выражен в радианах (градусов  $\times \pi/180$ ). Преобразование между градусами и радианами более подробно разъяснено в разделах, посвященных командам RAD и DEG.

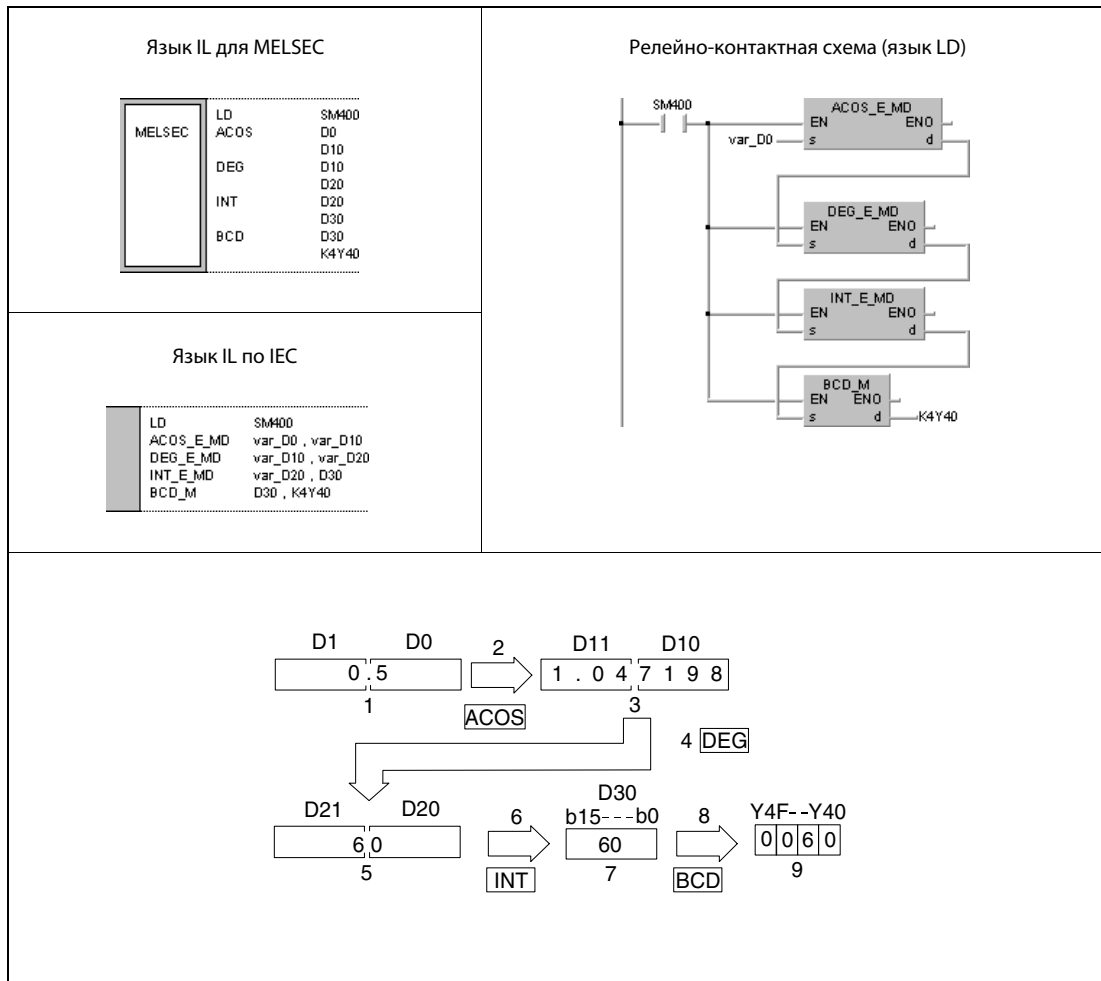
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Значение в s и s + 1 находится вне диапазона от -1 до 1 (код ошибки 4100).
- Если применяется процессор Q4AR и при этом указанный в s адрес содержит -0 и специальный маркер SM707 не установлен, выводится код ошибки 4100.

**Пример** ACOS

Следующая программа при включенном маркере SM400 вычисляет арккосинус числа с плавающей запятой (данных типа REAL), хранящегося в D0 и D1, и выводит полученный угол в радианах в виде 4-значного двоично-десятичного значения на выходы с Y20 по Y4F.



- 1 число с плавающей запятой (данные типа REAL)
- 2 вычисление арккосинуса
- 3 число с плавающей запятой (данные типа REAL)
- 4 преобразование угла
- 5 число с плавающей запятой (данные типа REAL)
- 6 преобразование в двоичный формат
- 7 двоичное значение
- 8 преобразование в двоично-десятичный формат
- 9 двоично-десятичное значение

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.12.6 ATAN, ATANP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

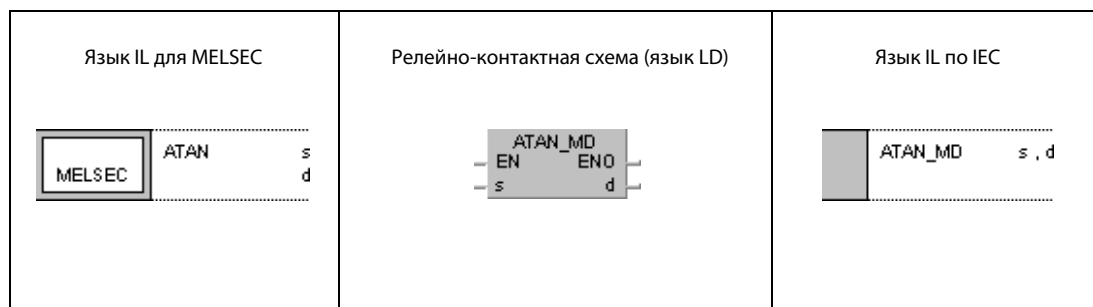
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

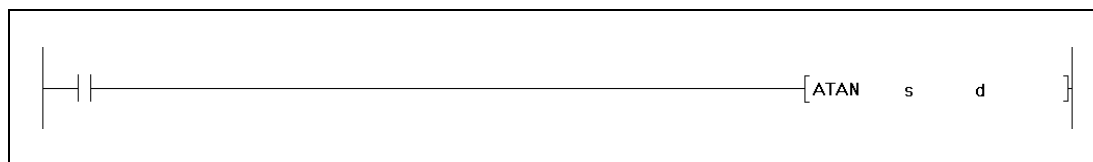
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы E			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	●	●	—	●	—	3	
d	—	●	●	—	●	●	—	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

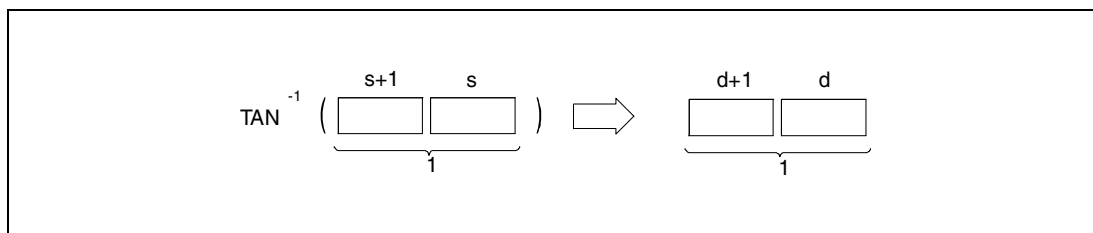
Операнд	Значение	Тип данных
s	Первый адрес операнда, содержащего тангенс, из которого вычисляется арктангенс.	REAL
d	Первый адрес операнда, в котором сохраняется результат.	

**Принцип действия**

**Вычисление арктангенса**

**ATAN Вычисление арктангенса числа с плавающей запятой**

Команда ATAN вычисляет угол по тангенсу, указанному в s и s + 1, и сохраняет результат в d и d + 1.



<sup>1</sup> число с плавающей запятой (данные типа REAL)

Угол в d и d + 1 выражен в радианах (градусов  $\times \pi/180$ ). Преобразование между градусами и радианами более подробно разъяснено в разделах, посвященных командам RAD и DEG.

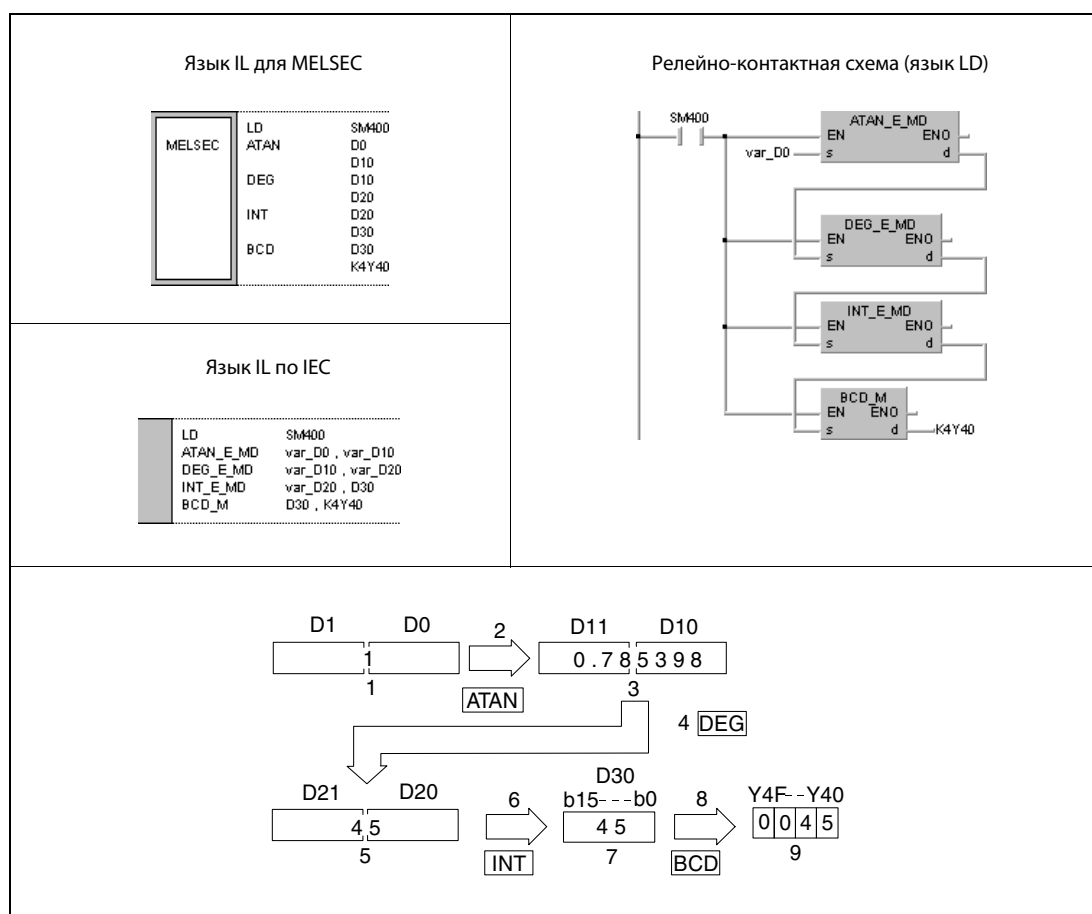
**Источники ошибок**

Если применяется процессор Q4AR и при этом указанный в s адрес содержит -0 и специальный маркер SM707 не установлен, выводится код ошибки 4100.



**Пример** ATAN

Следующая программа при включенном маркере SM400 вычисляет арктангенс числа с плавающей запятой (данных типа REAL), хранящегося в D0 и D1, и выводит полученный угол в радианах в виде 4-значного двоично-десятичного значения на выходы с Y20 по Y4F.



- 1 число с плавающей запятой (данные типа REAL)
- 2 вычисление арктангенса
- 3 число с плавающей запятой (данные типа REAL)
- 4 преобразование угла
- 5 число с плавающей запятой (данные типа REAL)
- 6 преобразование в двоичный формат
- 7 двоичное значение
- 8 преобразование в двоично-десятичный формат
- 9 двоично-десятичное значение

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

7.12.7 RAD, RADP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

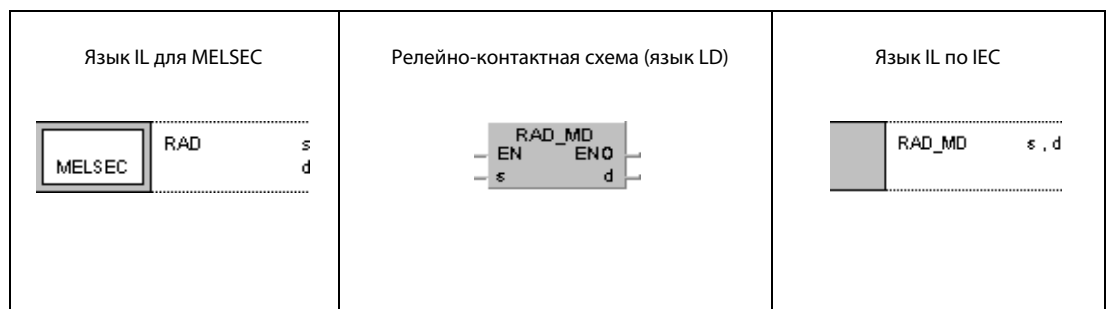
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

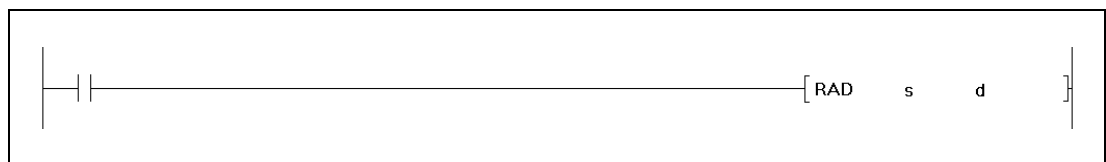
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы E			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	●	●	—	●	—	3	
d	—	●	●	—	●	●	—	—	—		

GX IEC Developer



GX Developer



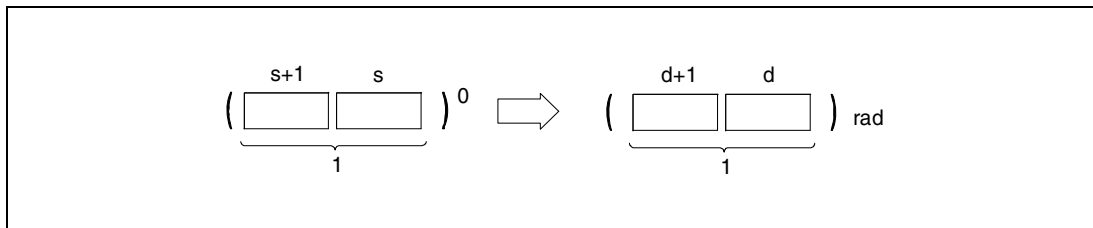
Переменные

Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранены угловые градусы, пересчитываемые в радианы.	REAL
d	Первый адрес операнда, в котором сохраняется результат преобразования.	

**Принцип действия**      **Пересчет из градусов в радианы**

**RAD**                      **Команда пересчета**

Команда RAD преобразует значение в градусах (°), указанное в s и s + 1, в значение в радианах (рад) и сохраняет результат в d и d + 1.



<sup>1</sup> число с плавающей запятой (данные типа REAL)

Пересчет из градусов в радианы происходит по следующей формуле:

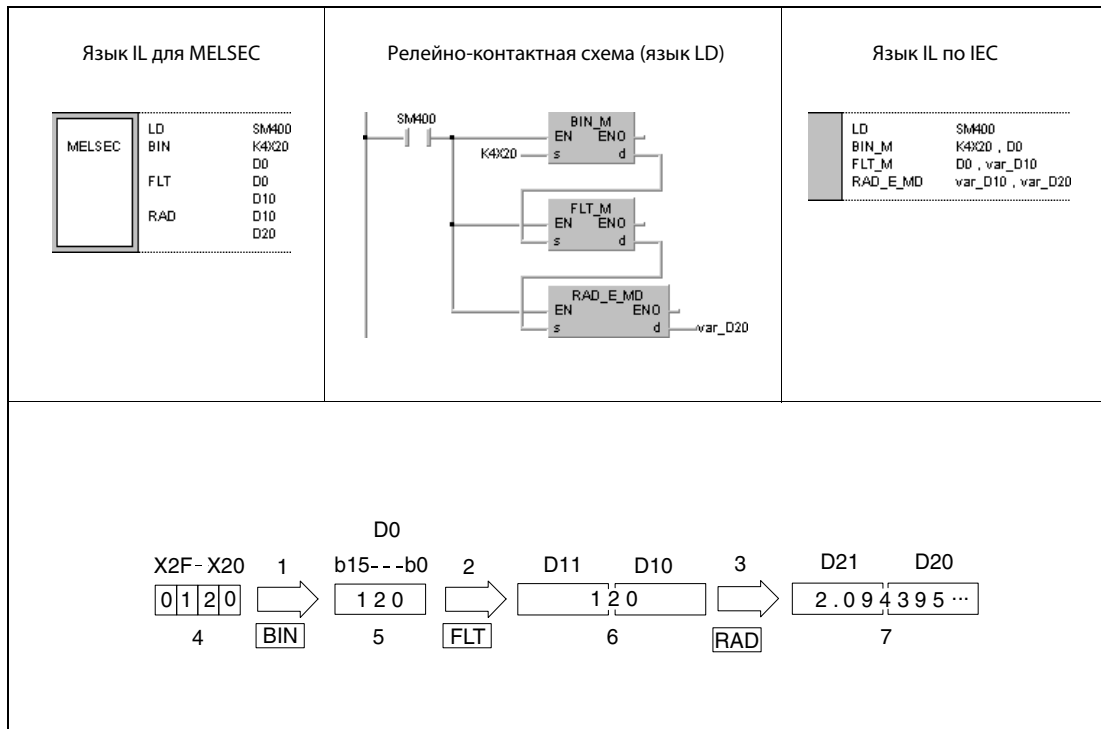
$$\text{значение в радианах} = \text{значение в градусах} \times \pi/180$$

**Источники ошибок**

Если применяется процессор Q4AR и при этом указанный в s адрес содержит -0 и специальный маркер SM707 не установлен, выводится код ошибки 4100.

**Пример** RAD

Следующая программа при включенном маркере SM400 преобразует значение в градусах, хранящееся в виде 4-значного двоично-десятичного значения в операндах с X20 по X2F, в радианы и сохраняет результат в виде числа с плавающей запятой в D20 и D21.



- 1 преобразование в двоичный формат
- 2 преобразование в формат с плавающей запятой
- 3 преобразование в радианы
- 4 двоично-десятичное значение
- 5 двоичное значение
- 6 число с плавающей запятой (данные типа REAL)
- 7 число с плавающей запятой (данные типа REAL)

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.12.8 DEG, DEGP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

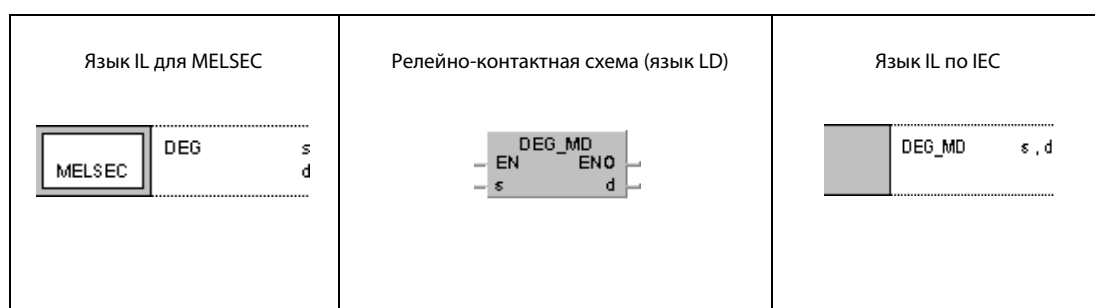
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

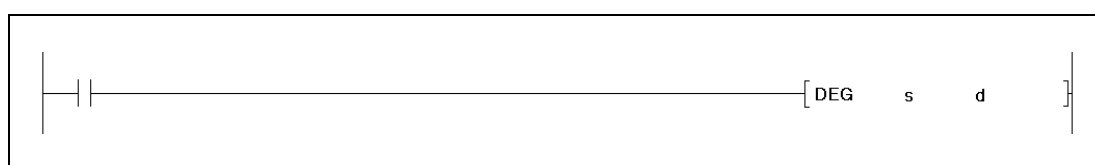
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы E			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	●	●	—	●	—	3	
d	—	●	●	—	●	●	—	—	—		

**GX IEC Developer**



**GX Developer**

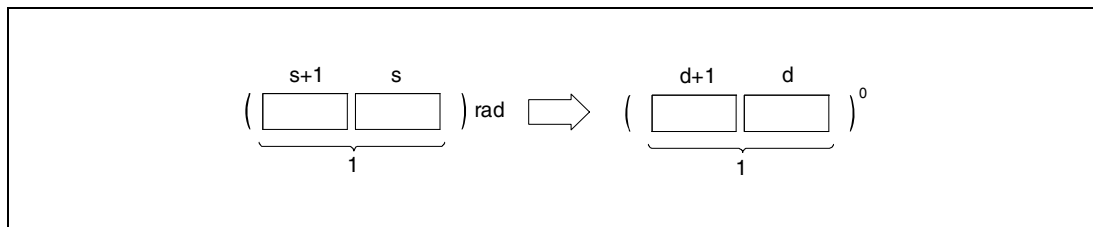


**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранено значение в радианах, преобразуемое в градусы.	REAL
d	Первый адрес операнда, в котором сохраняется результат преобразования.	

**Принцип действия****Пересчет из радиан в градусы****DEG Команда пересчета**

Команда DEG преобразует значение в радианах (рад), хранящееся в s и s + 1, в градусы (°) и сохраняет результат в d и d + 1.



<sup>1</sup> число с плавающей запятой (данные типа REAL)

Пересчет из радиан в градусы происходит по следующей формуле:

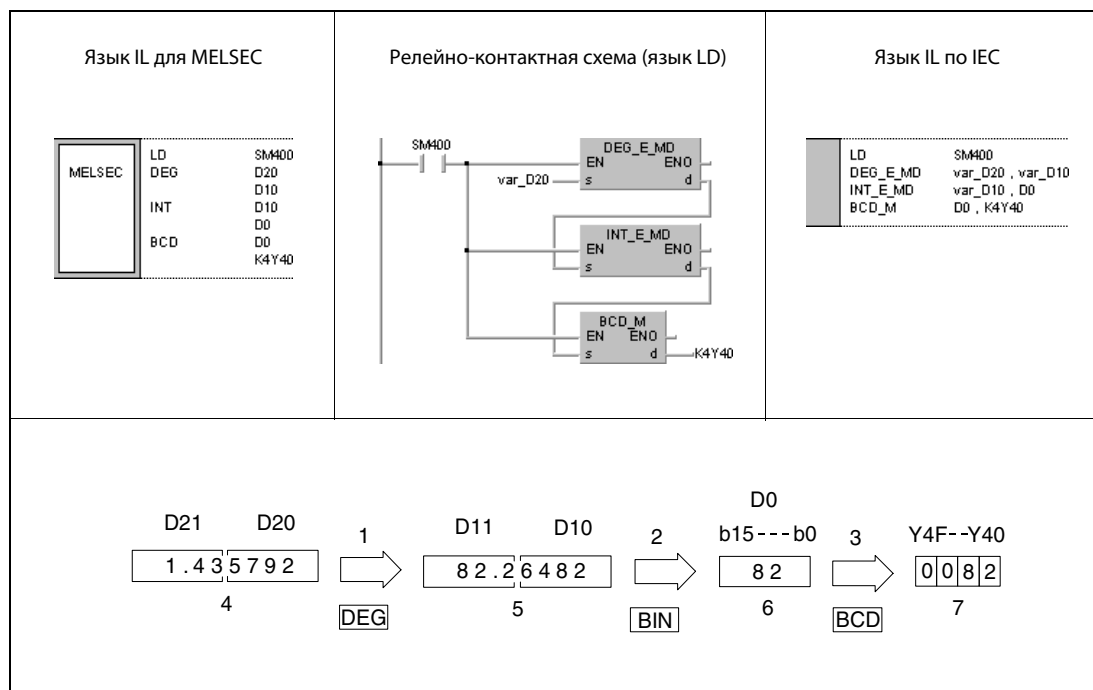
*значение в градусах = значение в радианах × π/180*

**Источники ошибок**

Если применяется процессор Q4AR и при этом указанный в s адрес содержит -0 и специальный маркер SM707 не установлен, выводится код ошибки 4100.

**Пример** DEG

Следующая программа при включенном маркере SM400 преобразует значение в радианах, хранящееся в D20 и D21 в 4-значном двоично-десятичном виде, в градусы и выводит результат в виде 4-значного двоично-десятичного числа на выходы с Y20 по Y4F.



- 1 преобразование в градусы
- 2 преобразование в двоичный формат
- 3 преобразование в двоично-десятичный формат
- 4 число с плавающей запятой (данные типа REAL)
- 5 число с плавающей запятой (данные типа REAL)
- 6 двоичное значение
- 7 двоично-десятичное значение

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.12.9 SQR, SQRP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

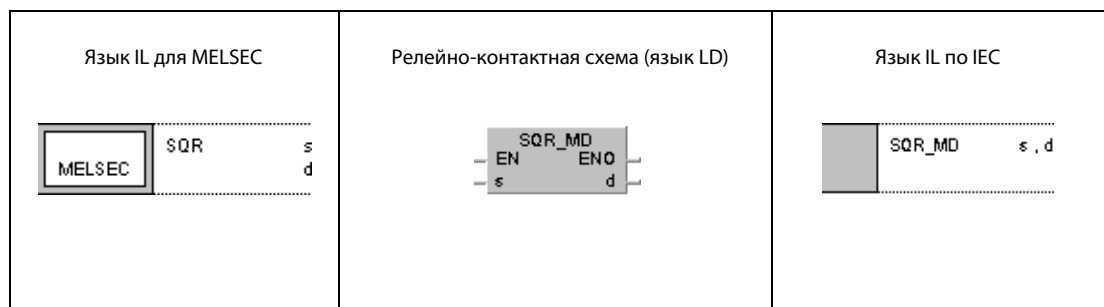
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

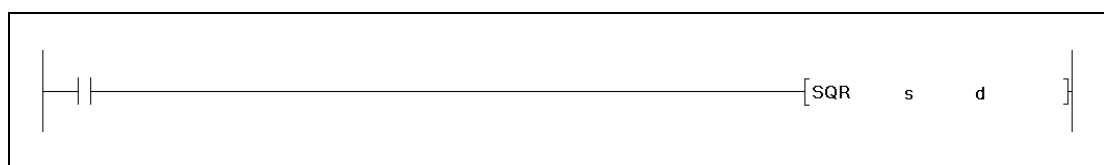
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непос. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы E			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	●	●	—	●	—	SM0	3
d	—	●	●	—	●	●	—	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда, содержащего значение, из которого извлекается квадратный корень.	REAL
d	Первый адрес операнда, в котором сохраняется результат вычисления.	

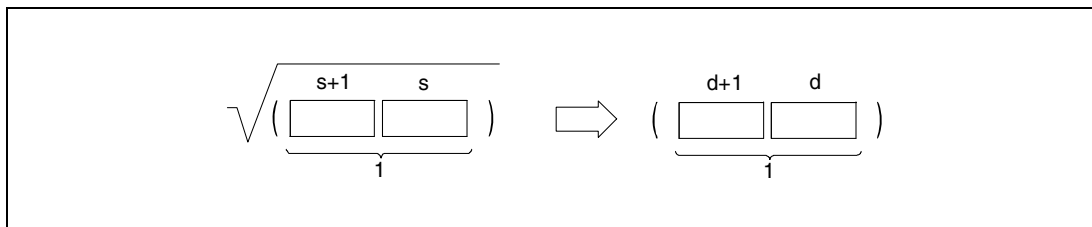


**Принцип действия**

**Вычисление квадратного корня**

**SQR Извлечение квадратного корня числа с плавающей запятой**

Команда SQR извлекает квадратный корень из числа с плавающей запятой, хранящегося в s и s + 1, и сохраняет результат в d и d + 1.



<sup>1</sup> число с плавающей запятой (данные типа REAL)

В s и s + 1 должны находиться только положительные значения. Обработка отрицательных значений не возможна.

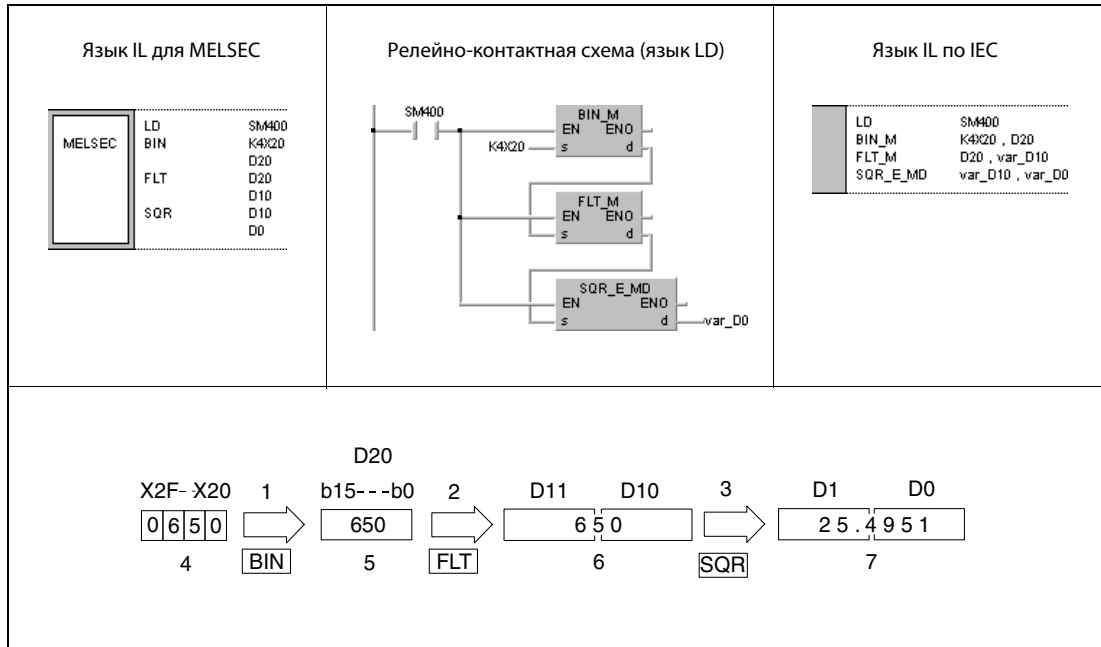
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Значение, хранящееся начиная с s, является отрицательным.
- Если применяется процессор Q4AR и при этом указанный в s адрес содержит -0 и специальный маркер SM707 не установлен, выводится код ошибки 4100.

**Пример** SQR

Следующая программа при включенном маркере SM400 извлекает квадратный корень из 4-значного двоично-десятичного значения, хранящегося в операндах с X20 по X2F, и сохраняет результат в виде числа с плавающей запятой в D0 и D1.



- 1 преобразование в двоичный формат
- 2 преобразование в формат с плавающей запятой
- 3 вычисление квадратного корня
- 4 двоично-десятичное значение
- 5 двоичное значение
- 6 число с плавающей запятой (данные типа REAL)
- 7 число с плавающей запятой (данные типа REAL)

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.12.10 EXP, EXPP

#### Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

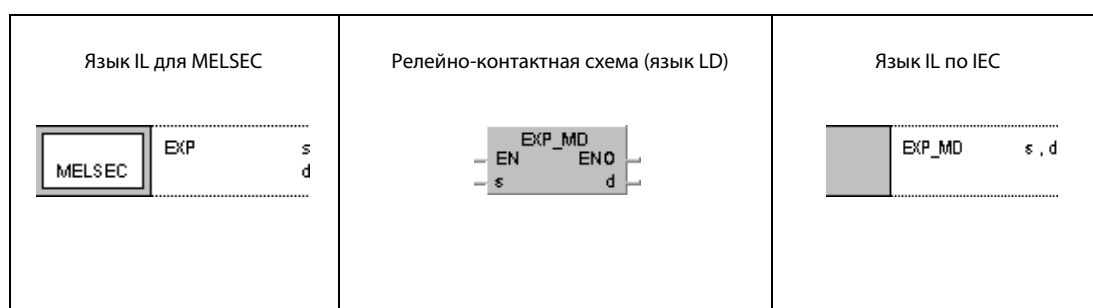
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

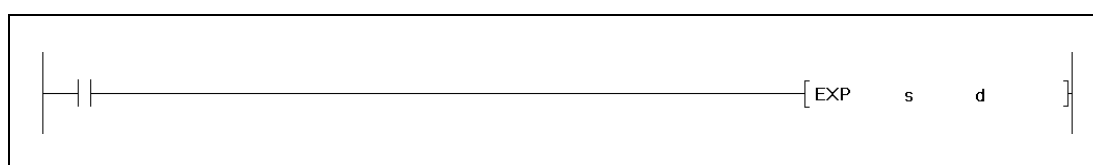
#### Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы E			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	●	●	—	●	—	SM0	3
d	—	●	●	—	●	●	—	—	—		

#### GX IEC Developer



#### GX Developer

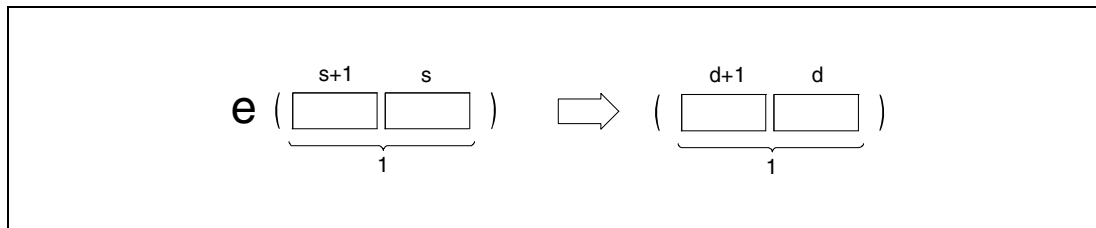


#### Переменные

Операнд	Значение	Тип данных
s	Первый адрес операнда, содержащего значение, в отношении которого выполняется команда EXP.	REAL
d	Первый адрес операнда, в котором сохраняется результат вычисления.	

**Принцип действия****Вычисление экспоненты****EXP                    Степень с основанием e**

Команда EXP возводит число e в степень, указанную в виде числа с плавающей запятой в s и s + 1, и сохраняет результат в d и d + 1.



<sup>1</sup> число с плавающей запятой (данные типа REAL)

Вычисление основывается на числе Эйлера (e = 2.718281828).

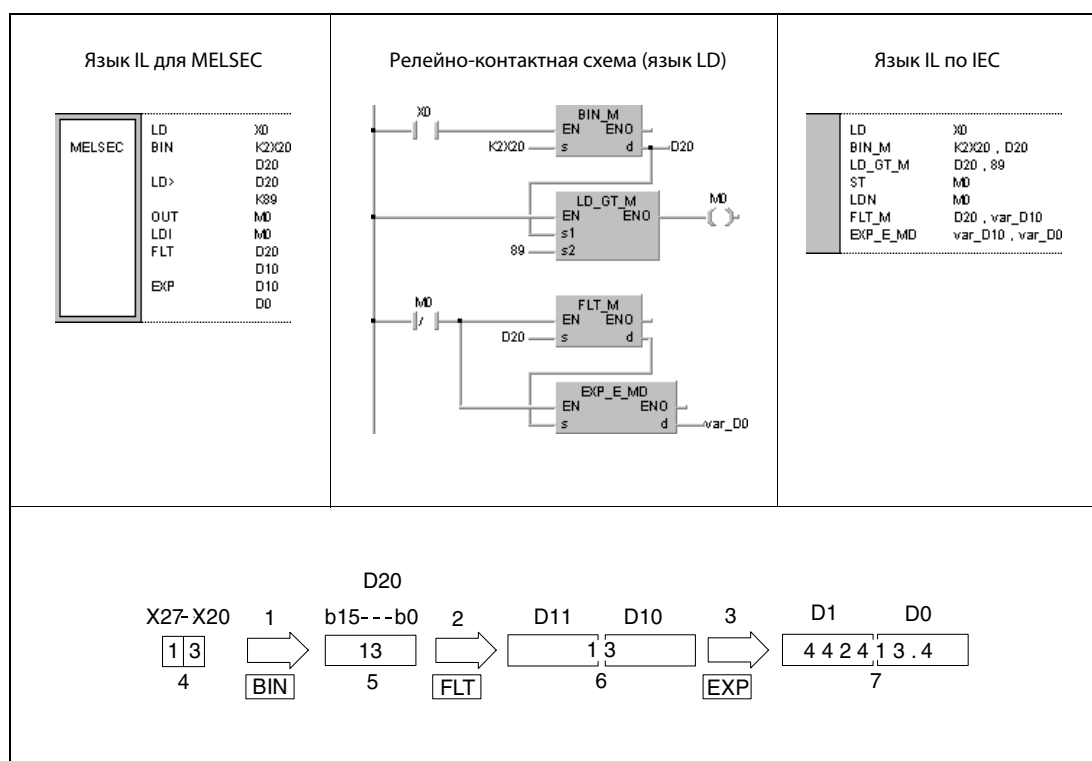
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Результат вычисления находится вне диапазона от  $2^{-127}$  до  $2^{129}$  (код ошибки 4100).
- Если применяется процессор Q4AR и при этом указанный в s адрес содержит -0 и специальный маркер SM707 не установлен, выводится код ошибки 4100.

**Пример** EXP

Следующая программа возводит число *e* в степень, указанную в виде 2-значного двоично-десятичного значения в операндах с X20 по X27, и сохраняет результат в виде числа с плавающей запятой в D0 и D1.



- 1 преобразование в двоичный формат
- 2 преобразование в формат с плавающей запятой
- 3 возведение в степень
- 4 двоично-десятичное значение
- 5 двоичное значение
- 6 число с плавающей запятой (данные типа REAL)
- 7 число с плавающей запятой (данные типа REAL)

**ПРИМЕЧАНИЯ** Результат вычисления не должен превышать  $2^{129}$ . Поэтому, если двоично-десятичное значение превышает 89 (так как  $\ln 2^{129} = \text{прибл. } 89.41598$ ), выработывается сообщение об ошибке SMO.

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.12.11 LOG, LOGP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

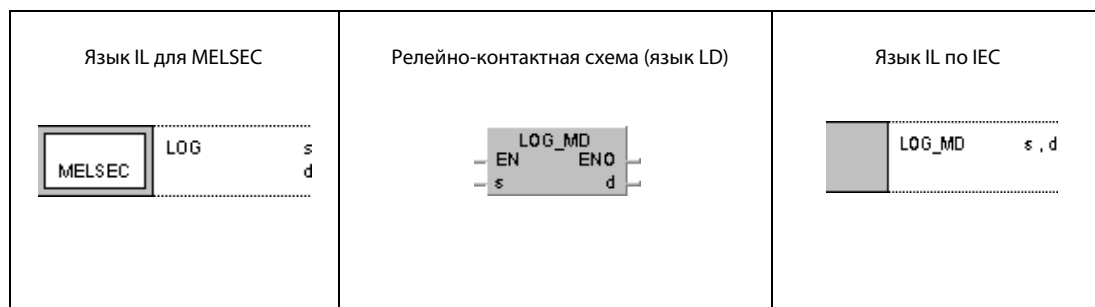
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

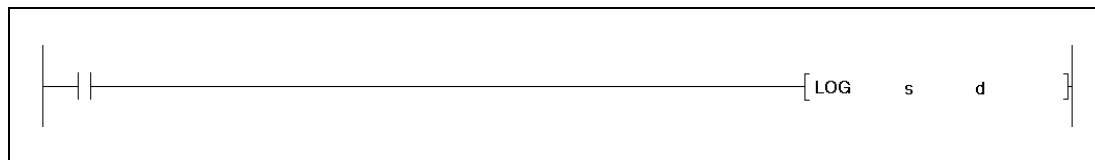
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непос. адр. J□□		Спец. модули U□V□□	Индексные регистры Zn	Константы E			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	●	●	—	●	—	SM0	3
d	—	●	●	—	●	●	—	—	—		

**GX IEC Developer**



**GX Developer**

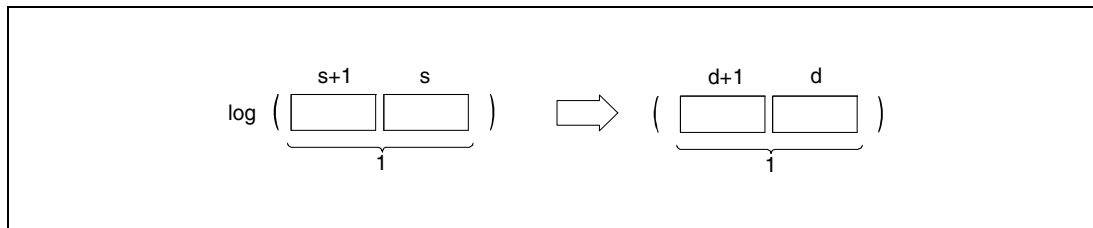


**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранено логарифмируемое значение.	REAL
d	Первый адрес операнда, в котором сохраняется результат вычисления.	

**Принцип действия****Вычисление натурального логарифма****LOG****Вычисление натурального логарифма числа с плавающей запятой**

Команда LOG вычисляет натуральный логарифм числа с плавающей запятой в  $s$  и  $s + 1$  и сохраняет результат в  $d$  и  $d + 1$ .



<sup>1</sup> число с плавающей запятой (данные типа REAL)

В  $s$  и  $s + 1$  можно указывать только положительные значения. Для отрицательных значений команда не выполняется.

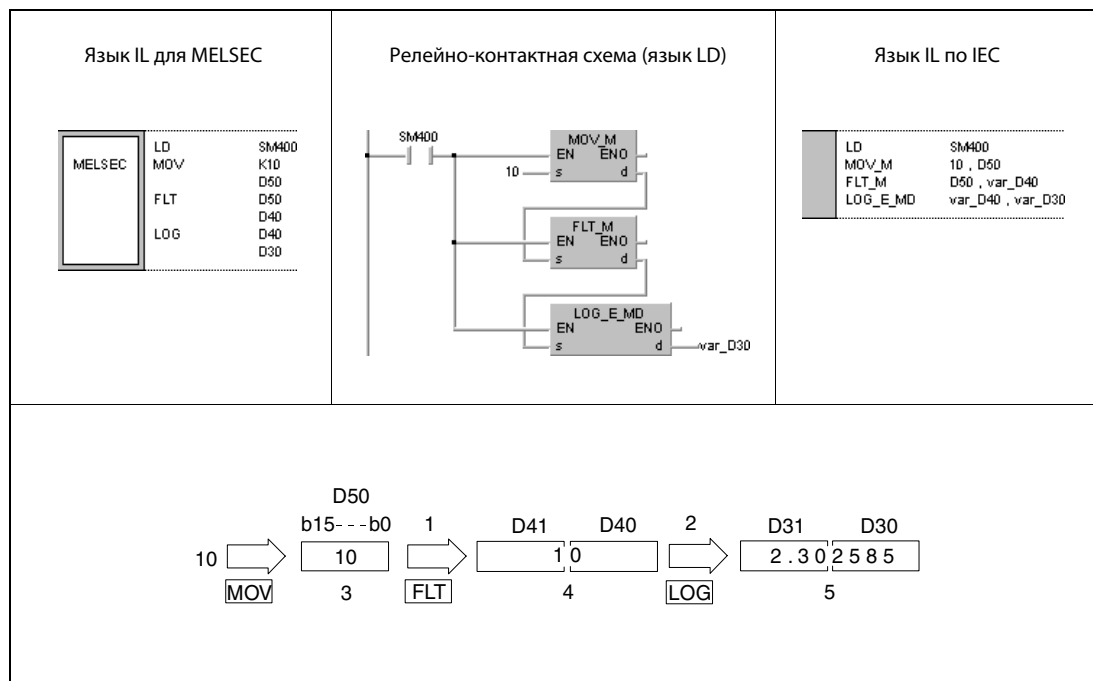
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В  $s$  указано отрицательное значение (код ошибки 4100).
- Результат вычисления находится вне диапазона от  $2^{-127}$  до  $2^{129}$  (код ошибки 4100).
- Если применяется процессор Q4AR и при этом указанный в  $s$  адрес содержит  $-0$  и специальный маркер SM707 не установлен, выводится код ошибки 4100.

**Пример 1** LOG

Следующая программа вычисляет натуральный логарифм значения 10 и сохраняет результат по адресу D30 и D31.



- 1 преобразование в формат с плавающей запятой
- 2 вычисление логарифма
- 3 двоичное значение
- 4 число с плавающей запятой (данные типа REAL)
- 5 число с плавающей запятой (данные типа REAL)

**ПРИМЕЧАНИЕ** *Центральный процессор работает с натуральными логарифмами (логарифмами с основанием e). Для пересчета в десятичный логарифм (логарифм с основанием 10) можно воспользоваться следующей формулой:*

$$\log_{10} X = 0.43429 \times \log_e X$$

**ПРИМЕЧАНИЕ** *Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.*



### 7.12.12 RND, RNDP, SRND, SRNDP

**Процессор**

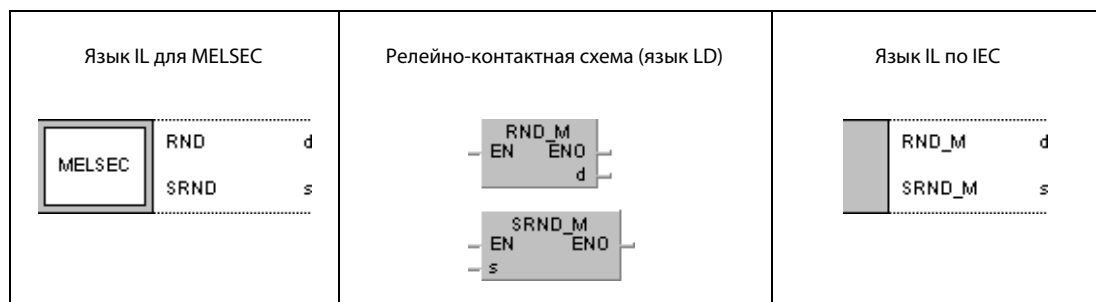
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

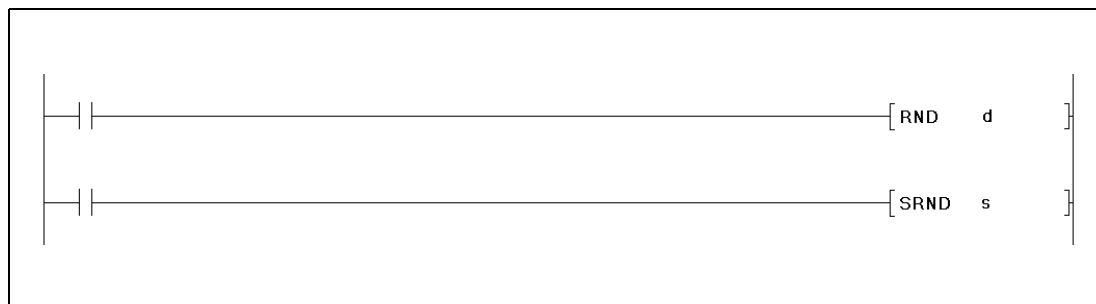
**Операнды MELSEC Q**

Операнды								Флаг ошибки	Шагов	
Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
битовые	словные		битовые	словные						
s					●		—	—	2	

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
d	Первый адрес операнда, в котором сохраняется случайное число.	BIN, 16 бит
s	Серия случайных чисел или первый адрес операнда, в котором сохранены эти данные.	

**Принцип действия**      **Генерирование случайных чисел и обновление серии**

**RND**                      **Генерирование случайных чисел**

Команда RND вырабатывает случайное число в диапазоне от 0 до 32767 и сохраняет его в d.

**SRND**                      **Обновление серии случайных чисел**

Команда SRND обновляет серию случайных чисел, сохраненную в s.

**Пример 1**      RND

Следующая программа при включенном входе X10 вырабатывает случайное число и сохраняет его в D100.



**Пример 2**      SRND

Следующая программа при включенном входе X10 обновляет серию случайных чисел в D0.



### 7.12.13 BSQR, BSQRP, BDSQR, BDSQRP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

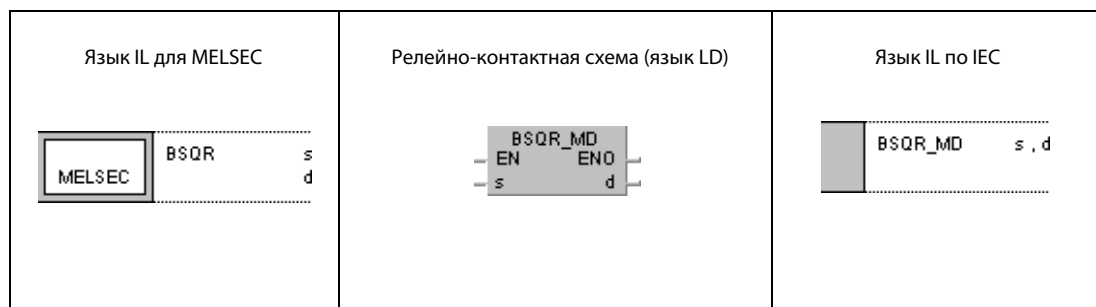
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

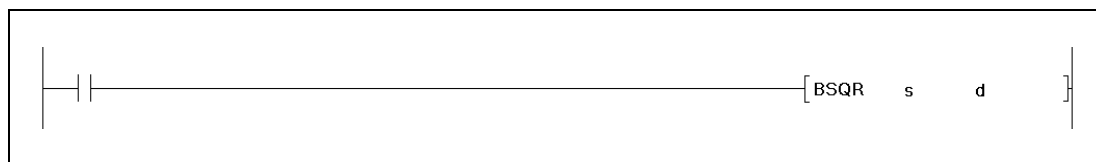
**Операнды MELSEC Q**

	Операнды							Флаг ошибки	Шагов		
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□□	Индексные регистры Zn			Константы K, H (16#)	Иные
	битовые	словные		битовые	словные						
s			●				●	—	SM0	3	
d			●				—	—			

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Данные, из которых извлекается квадратный корень, или первый адрес операнда, в котором эти данные сохранены.	BCD, 4/8 знаков
d	Первый адрес операнда, в котором сохраняется квадратный корень.	BCD, 4 знака

**Принцип действия****Вычисление квадратного корня из 4/8-значных двоично-десятичных данных****BSQR            Вычисление квадратного корня из 4-значного двоично-десятичного числа**

Команда BSQR вычисляет квадратный корень из  $s$  и сохраняет результат в  $d$  и  $d + 1$ .

$$\sqrt{s} = \boxed{\text{d}}_1 \cdot \boxed{\text{d+1}}_2$$

<sup>1</sup> целочисленная часть  
<sup>2</sup> дробная часть

Значение в  $s$  может быть только двоично-десятичным числом, имеющим максимум 4 знака. Это число должно находиться в диапазоне от 0 до 9999.

Результат вычисления, сохраняемый в  $d$  и  $d + 1$ , не должен выходить из диапазона от 0 до 9999.

Результат вычисляется с точностью до 5 знаков и округляется до 4-значного числа.

**BDSQR            Извлечение квадратного корня из 8-значного двоично-десятичного числа**

Команда BDSQR извлекает квадратный корень из  $s$  и  $s + 1$  и сохраняет результат в  $d$  и  $d + 1$ .

$$\sqrt{\underbrace{\boxed{s+1} \ \boxed{s}}_1} = \boxed{\text{d}}_2 \ \boxed{\text{d+1}}_3$$

<sup>1</sup> двухсловные данные  
<sup>2</sup> целочисленная часть  
<sup>3</sup> дробная часть

Значение в  $s$  и  $s + 1$  может быть только двоично-десятичным значением, содержащим максимум 8 знаков. Это число должно находиться в диапазоне от 0 до 99999999.

Результат вычисления, сохраняемый в  $d$  и  $d + 1$ , не должен выходить из диапазона от 0 до 9999.

Результат вычисляется с точностью до 5 знаков и округляется до 4-значного числа.

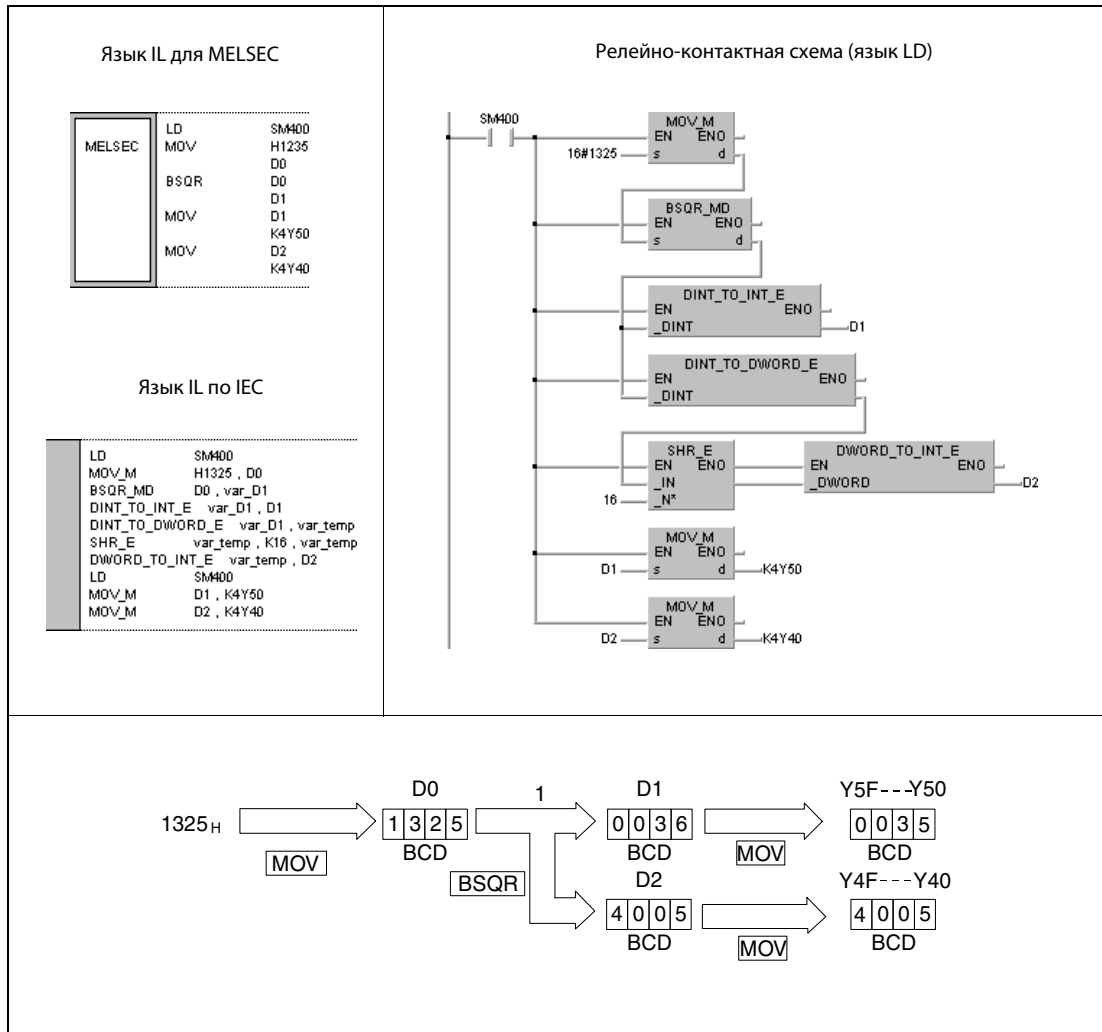
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Данные, сохраненные, начиная с  $s$ , не являются двоично-десятичными (код ошибки 4100).

**Пример 1** BSQR

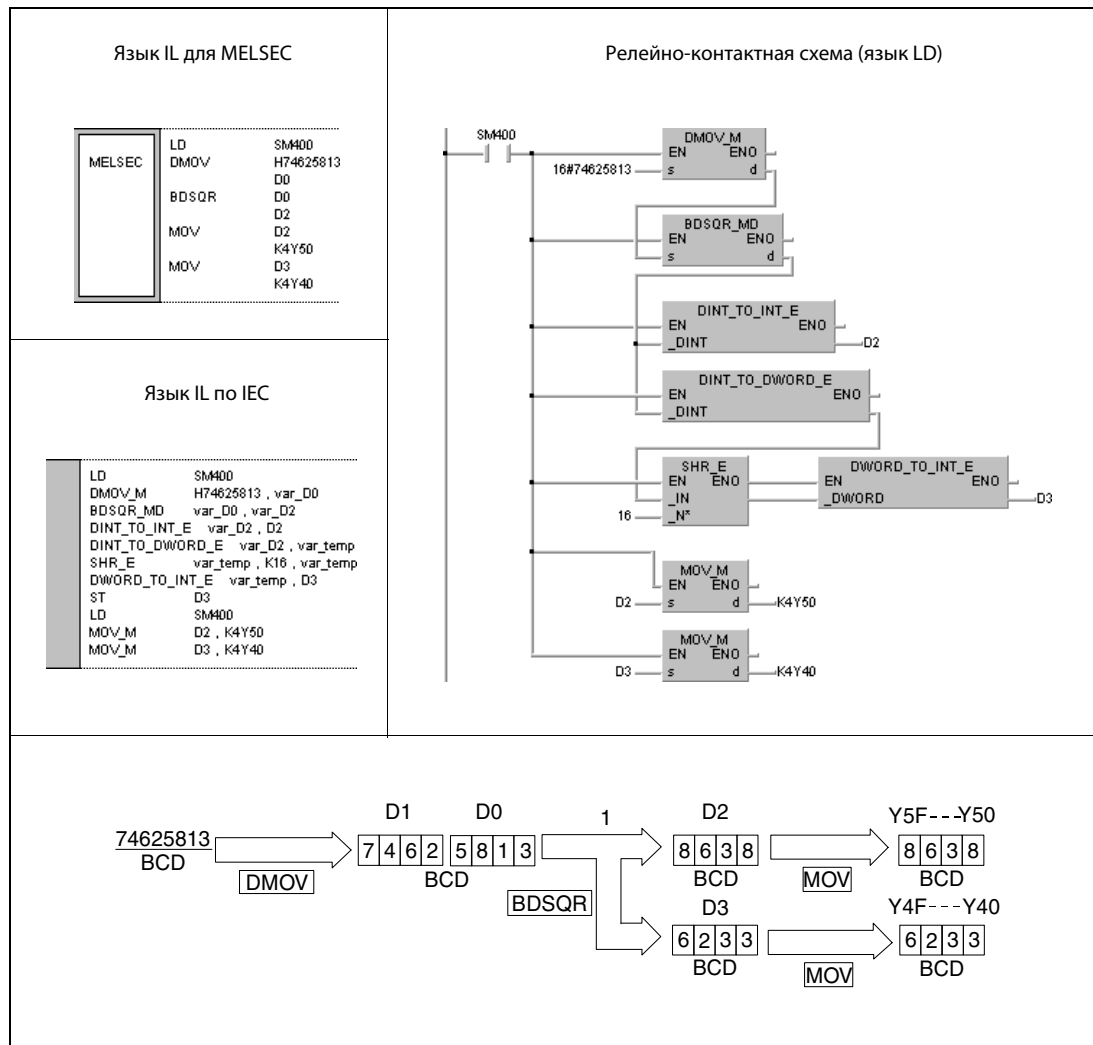
Следующая программа при включенном маркере SM400 вычисляет квадратный корень двоично-десятичного значения 1325 и выводит целочисленную часть результата в виде 4-значного двоично-десятичного значения на выходы с Y50 по Y5F, а дробную часть – в виде 4-значного двоично-десятичного значения на выходы с Y40 по Y4F.



<sup>1</sup> вычисление квадратного корня

## Пример 2 BDSQR

Следующая программа при включенном маркере SM400 вычисляет квадратный корень двоично-десятичного значения 74625813 и выводит целочисленную часть результата в виде 4-значного двоично-десятичного значения на выходы с Y50 по Y5F, а дробную часть – в виде 4-значного двоично-десятичного значения на выходы с Y40 по Y4F.



<sup>1</sup> вычисление квадратного корня

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.12.14 BSIN, BSINP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

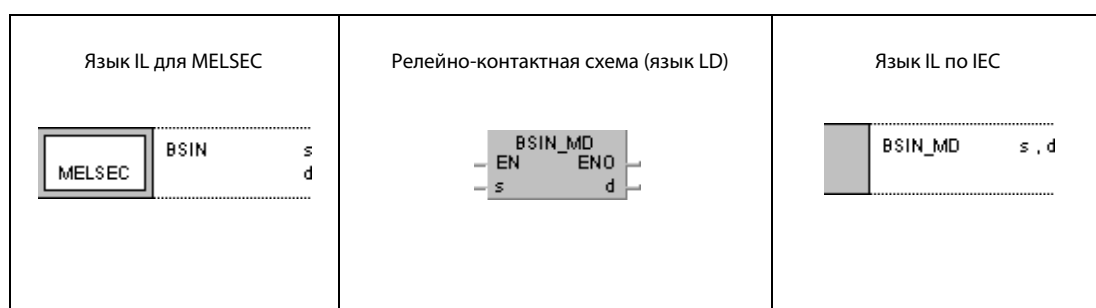
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

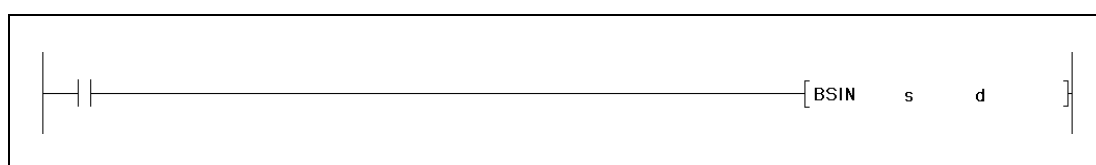
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	—	SM0	3	
d	—	●	●	—	—	—	—	—			

**GX IEC Developer**



**GX Developer**

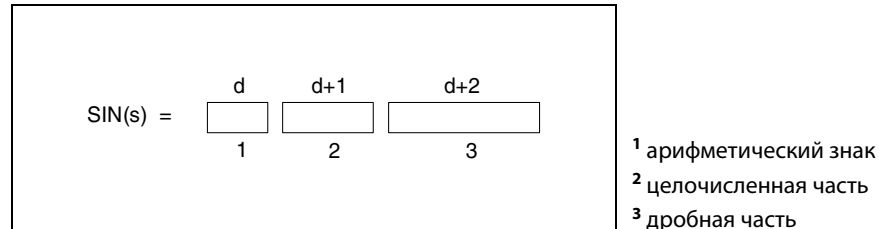


**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранены данные угла, для которого вычисляется синус.	BCD, 4 знака
d	Первый адрес операнда, в котором сохраняется результат.	

**Принцип действия****Вычисление синуса двоично-десятичных данных****BSIN                    Вычисление синуса**

Команда BSIN вычисляет синус угла, указанного в s. Арифметический знак результата сохраняется в d. Числовое значение результата сохраняется в d + 1 и d + 2.



В s должно быть указано двоично-десятичное значение в диапазоне от 0° до 360°.

При положительном значении арифметический знак результата в d принимает значение 0, а при отрицательном значении – значение 1.

Результат в d + 1 и d + 2 может представлять собой двоично-десятичное значение в диапазоне между –1.000 и 1.000.

Результат вычисления округляется, начиная с 5-го разряда.

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанные в s данные не являются двоично-десятичными (код ошибки 4100).
- Указанные в s данные находятся вне диапазона от 0° до 360° (код ошибки 4100).

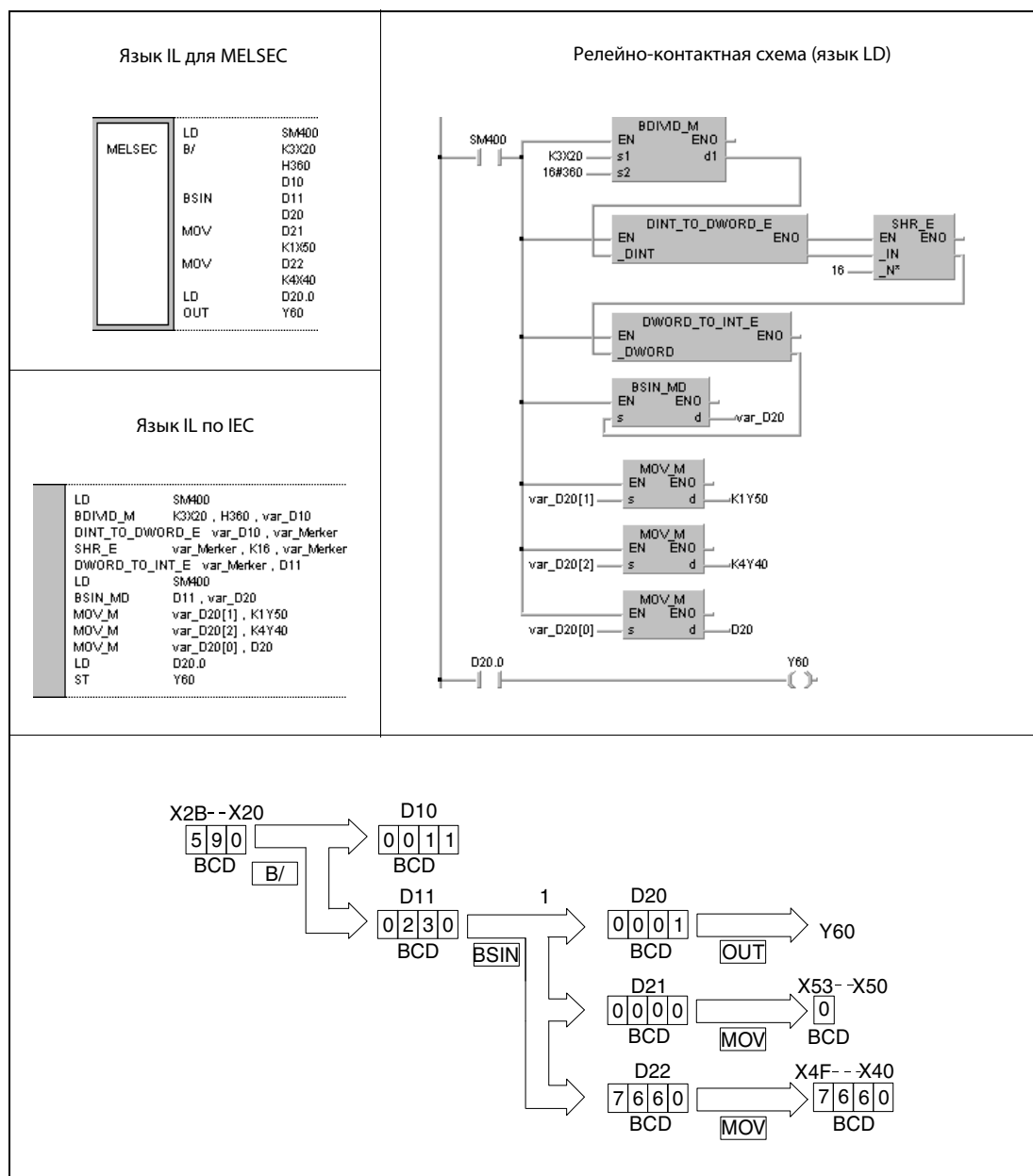


**Пример** BSIN

Следующая программа при включенном маркере SM400 вычисляет синус 3-значного двоично-десятичного числа на входах с X20 по X2B. Если значение по адресу с X20 по X2B больше 360, оно корректируется на диапазон между 0° и 360°.

Арифметический знак выводится на Y60. Целочисленная часть выводится в виде 1-значного двоично-десятичного значения на выходы Y50...Y53.

Дробная часть выводится в виде 4-значного двоично-десятичного значения на выходы с Y40 по Y4F.



<sup>1</sup> вычисление синуса

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

7.12.15 BCOS, BCOSP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

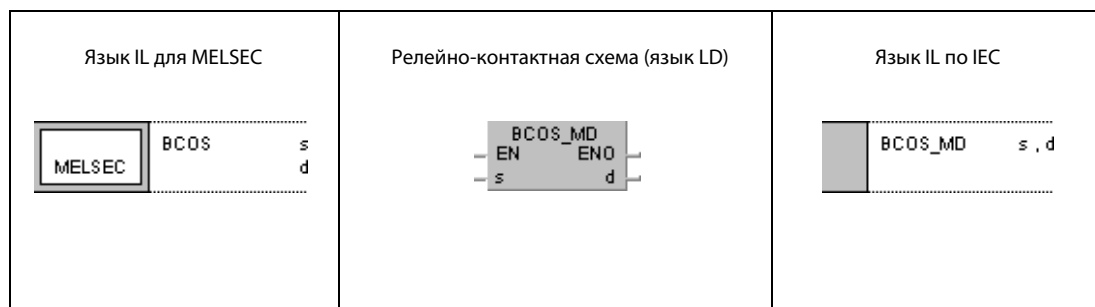
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

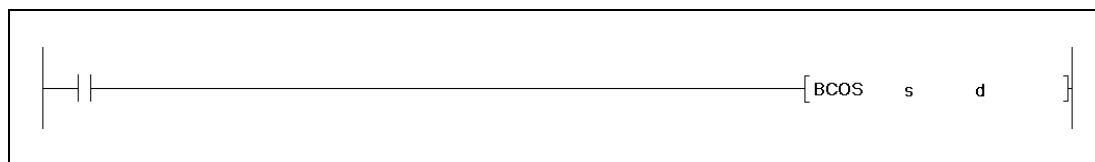
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	●	—	SM0	3
d	—	●	●	—	—	—	—	—	—		

GX IEC Developer



GX Developer

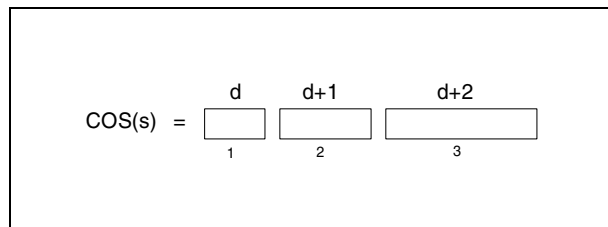


Переменные

Операнд	Значение	Тип данных
s	Первый адрес операнда со значением угла, в отношении которого выполняется команда BCOS (косинус).	BCD, 4 знака
d	Первый адрес операнда, в котором сохраняется результат.	

**Принцип действия****Вычисление косинуса двоично-десятичных данных****BCOS                      Вычисление косинуса**

Команда BCOS вычисляет косинус угла, указанного в s. Арифметический знак результата сохраняется в d. Числовое значение результата сохраняется в d + 1 и d + 2.



- <sup>1</sup> арифметический знак
- <sup>2</sup> целочисленная часть
- <sup>3</sup> дробная часть

Значение в s должно быть двоично-десятичным значением в диапазоне от 0° до 360°.

При положительном значении арифметический знак результата в d принимает значение 0, а при отрицательном значении – значение 1.

Результат в d + 1 и d + 2 может представлять собой двоично-десятичное значение в диапазоне между –1.000 и 1.000.

Результат вычисления округляется, начиная с 5-го разряда.

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанные в s данные не являются двоично-десятичными (код ошибки 4100).
- Указанные в s данные находятся вне диапазона от 0° до 360° (код ошибки 4100).

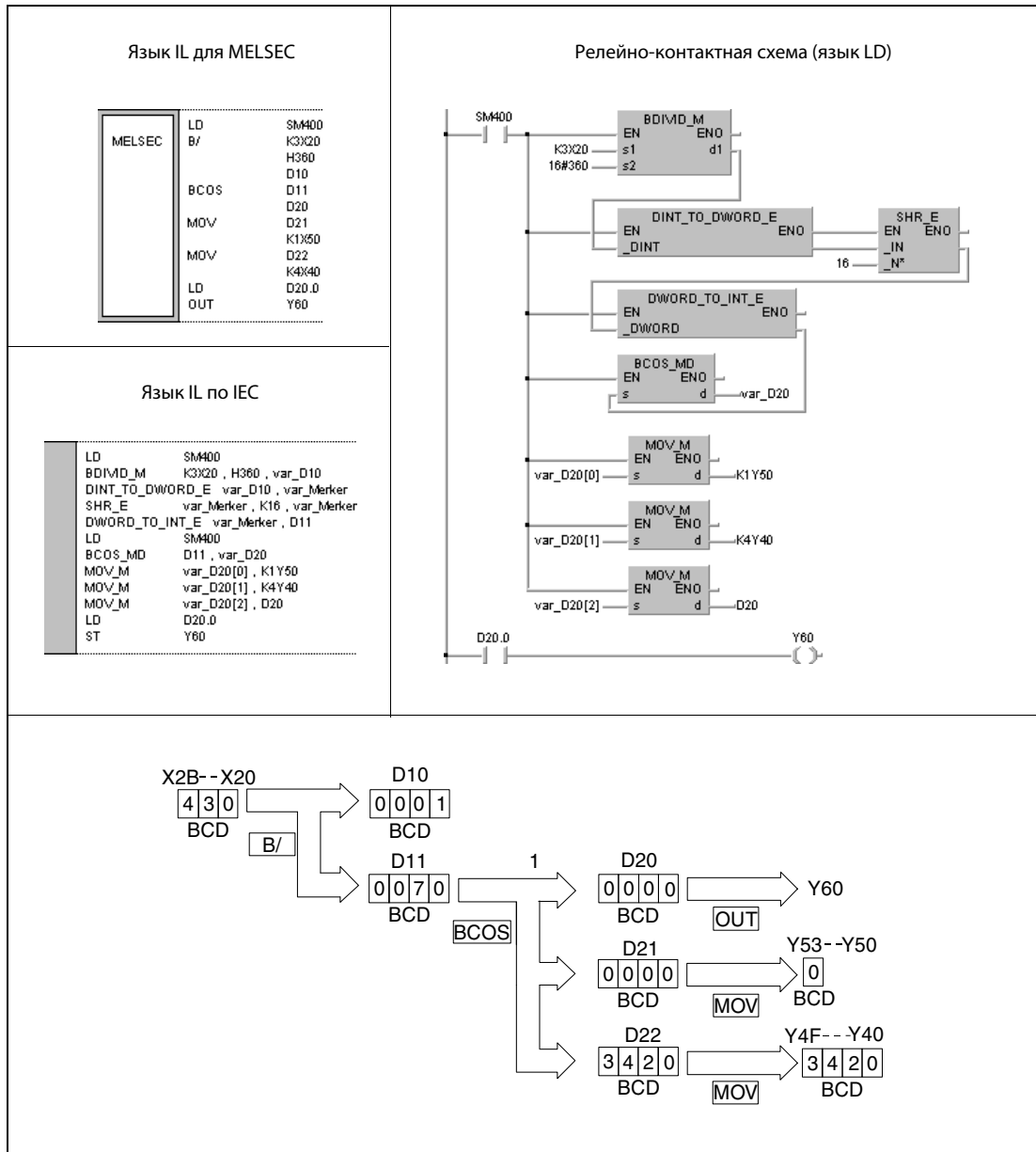
**Пример**

**BCOS**

Следующая программа при включенном маркере SM400 вычисляет косинус 3-значного двоично-десятичного значения, заданного на входах с X20 по X2B. Если значение по адресу с X20 по X2B больше 360, оно корректируется на диапазон между 0° и 360°.

Арифметический знак выводится на Y60. Целочисленная часть выводится в виде 1-значного двоично-десятичного значения на выходы Y50...Y53.

Дробная часть выводится в виде 4-значного двоично-десятичного значения на выходы с Y40 по Y4F.



<sup>1</sup> вычисление косинуса

**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.12.16 BTAN, BTANP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

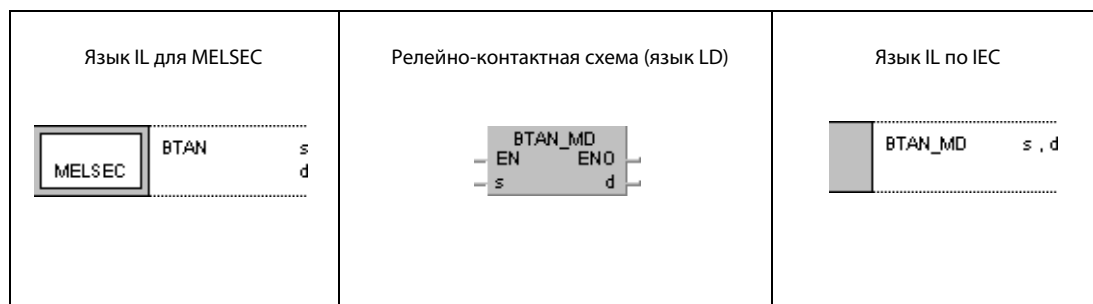
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

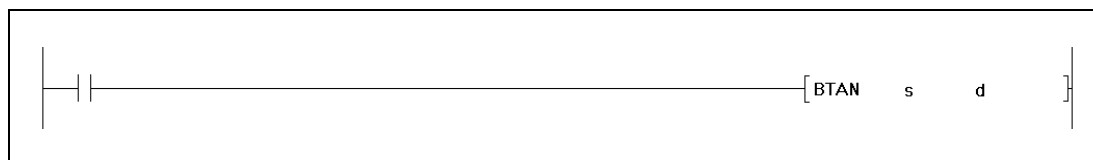
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	●	—	SM0	3
d	—	●	●	—	—	—	—	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда со значением угла, в отношении которого выполняется команда BTAN (тангенс).	BCD, 4 знака
d	Первый адрес операнда, в котором сохраняется результат.	

**Принцип действия****Вычисление тангенса двоично-десятичных данных****BTAN                      Вычисление тангенса**

Команда BTAN вычисляет тангенс угла, указанного в s. Арифметический знак результата сохраняется в d. Числовое значение результата сохраняется в d + 1 и d + 2.

$$\text{TAN}(s) = \boxed{\phantom{0}}_1 \boxed{\phantom{0}}_2 \cdot \boxed{\phantom{0}}_3$$

- <sup>1</sup> арифметический знак
- <sup>2</sup> целочисленная часть
- <sup>3</sup> дробная часть

Значение в s должно быть двоично-десятичным значением в диапазоне от 0° до 360°.

При положительном значении арифметический знак результата в d принимает значение 0, а при отрицательном значении – значение 1.

Результат в d + 1 и d + 2 может представлять собой двоично-десятичное значение в диапазоне между –57.2900 и 57.2900.

Результат вычисления округляется, начиная с 5-го разряда.

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

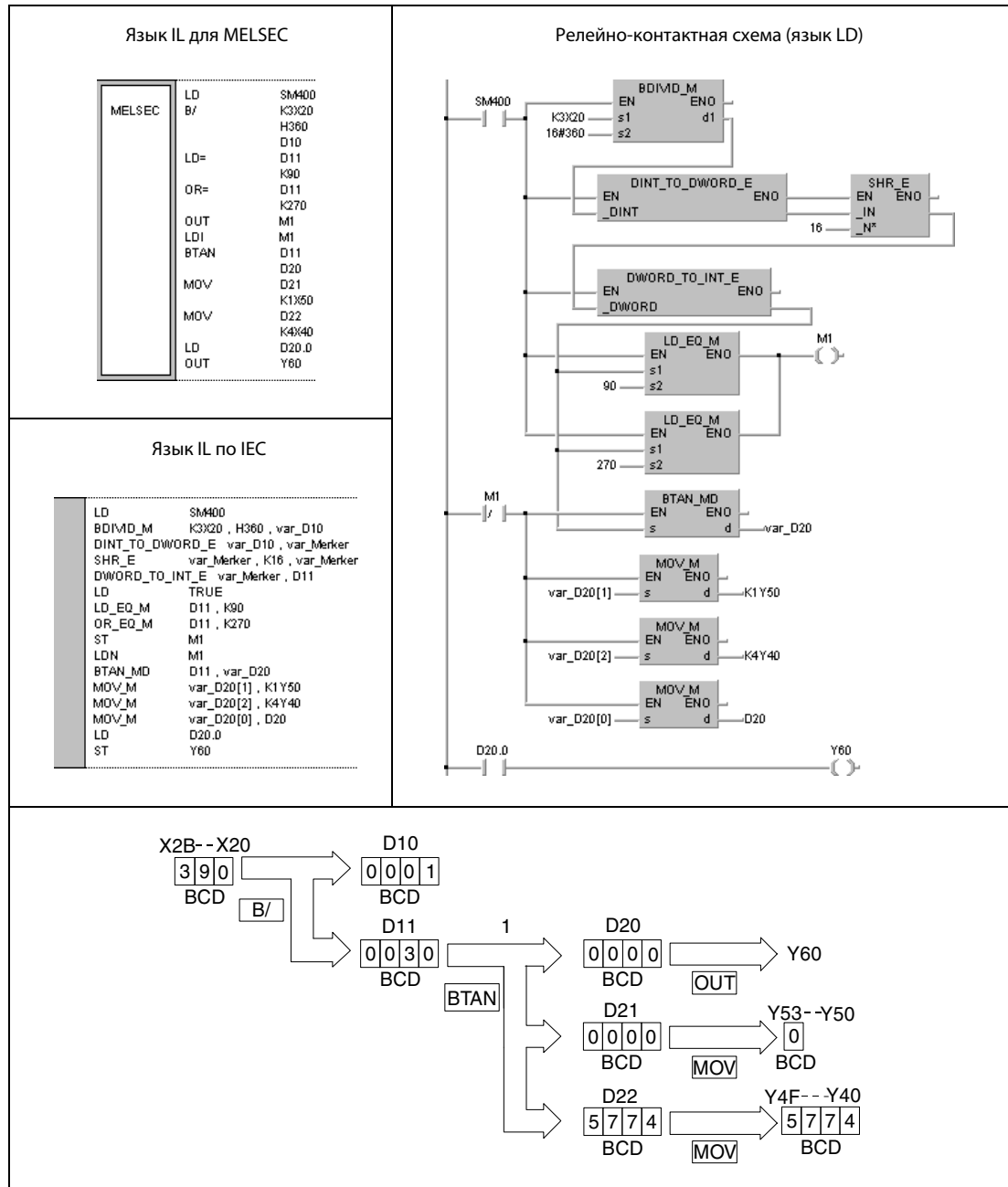
- Указанные в s данные не являются двоично-десятичными (код ошибки 4100).
- Указанные в s данные находятся вне диапазона от 0° до 360° (код ошибки 4100).
- В s указано значение 90° или 270° (код ошибки 4100).

**Пример** BTAN

Следующая программа при включенном маркере SM400 вычисляет тангенс 3-значного двоично-десятичного значения на входах с X20 по X2B. Если значение по адресу с X20 по X2B больше 360, оно корректируется на диапазон между 0° и 360°.

Арифметический знак выводится на выход Y60. Целочисленная часть выводится в виде 1-значного двоично-десятичного значения на выходы Y50...Y53.

Дробная часть выводится в виде 4-значного двоично-десятичного значения на выходы с Y40 по Y4F.



<sup>1</sup> вычисление тангенса

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

7.12.17 BASIN, BASINP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

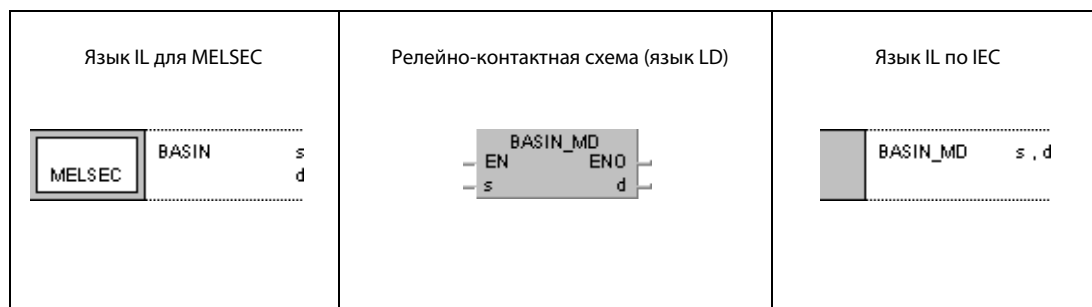
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

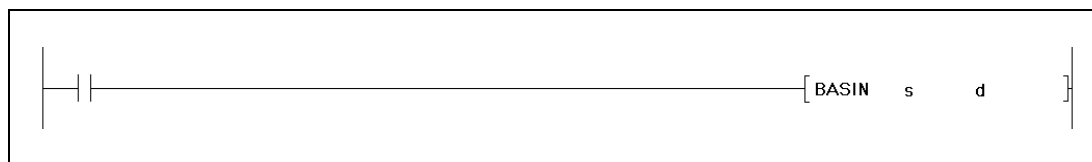
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
s	Первый адрес операнда со значением синуса, в отношении которого выполняется команда BASIN (арксинус).	BCD, 4 знака
d	Первый адрес операнда, в котором сохраняется результат.	



**Принцип действия****Вычисление арксинуса двоично-десятичных данных****BASIN          Вычисление арксинуса**

Команда BASIN вычисляет угол по синусу в  $s$ ,  $s + 1$  и  $s + 2$  и сохраняет результат в  $d$ .

$$\text{SIN}^{-1} = \left( \begin{array}{c} s \\ \boxed{\phantom{00}} \\ 1 \end{array} \begin{array}{c} s+1 \\ \boxed{\phantom{00}} \\ 2 \end{array} \cdot \begin{array}{c} s+2 \\ \boxed{\phantom{00}} \\ 3 \end{array} \right) = d$$

<sup>1</sup> арифметический знак

<sup>2</sup> целочисленная часть

<sup>3</sup> дробная часть

При положительном значении арифметический знак в  $s$  принимает значение 0, а при отрицательном значении – значение 1.

Целочисленной частью перед десятичной запятой и дробной частью после запятой должны быть только двоично-десятичные (BCD) значения в диапазоне между 0 и 1.0000.

Результат в  $d$  может быть двоично-десятичным значением в диапазоне между  $0^\circ$  и  $90^\circ$  или между  $270^\circ$  и  $360^\circ$ .

Результат вычисления округляется, начиная с 5-го десятичного разряда после запятой.

**Источники ошибок**

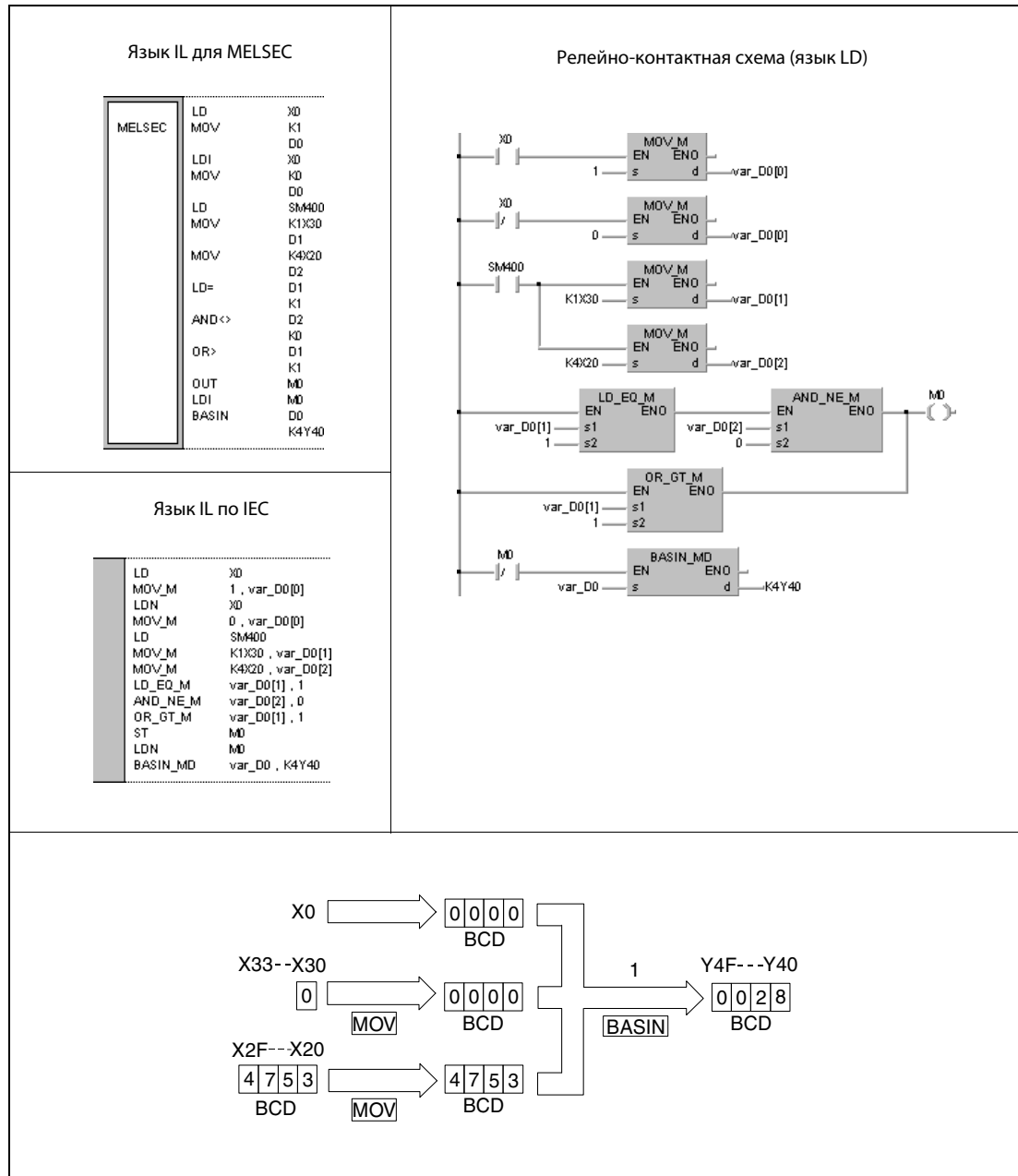
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Данные от  $s$  до  $s + 2$  не являются двоично-десятичными (код ошибки 4100).
- Данные, указанные от  $s$  до  $s + 2$ , находятся вне диапазона от  $-1.0000$  до  $1.0000$  (код ошибки 4100).

Пример

BASIN

Следующая программа при включенном маркере SM400 вычисляет арксинус данных, имеющих следующий состав: арифметический знак на входе X0 (положительный, если X0 установлен; отрицательный, если X0 не установлен), 1-значная целочисленная часть в формате BCD на входах X30...X33 и 4-значная дробная часть в формате BCD на входах X20...X2F. Вычисленное значение угла выводится в виде 4-значного двоично-десятичного значения на выходы с Y40 по Y4F.



<sup>1</sup> вычисление арксинуса

**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке ПОУ этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 7.12.18 BACOS, BACOSP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

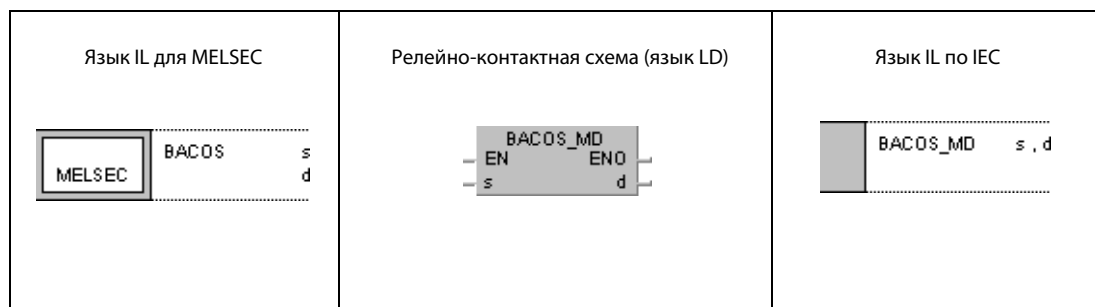
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

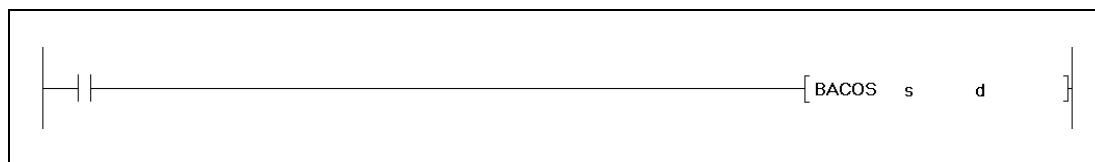
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда со значением косинуса, в отношении которого выполняется команда BACOS (арккосинус).	BCD, 4 знака
d	Первый адрес операнда, в котором сохраняется результат.	

**Принцип действия****Вычисление арккосинуса двоично-десятичных данных****BACOS      Вычисление арккосинуса**

Команда BACOS вычисляет угол по косинусу, указанному в  $s$ ,  $s + 1$  и  $s + 2$ , и сохраняет результат в  $d$ .

$$\text{COS}^{-1} = ( \boxed{\phantom{0}}_1^s \boxed{\phantom{0}}_2^{s+1} \cdot \boxed{\phantom{0}}_3^{s+2} ) = d$$

- <sup>1</sup> арифметический знак
- <sup>2</sup> целочисленная часть
- <sup>3</sup> дробная часть

При положительном значении арифметический знак в  $s$  принимает значение 0, а при отрицательном значении – значение 1.

Целочисленной частью перед десятичной запятой и дробной частью после запятой должны быть только двоично-десятичные (BCD) значения в диапазоне между 0 и 1.0000.

Результат в  $d$  представляет собой двоично-десятичное значение в диапазоне между  $0^\circ$  и  $180^\circ$ .

Результат вычисления округляется, начиная с 5-го десятичного разряда после запятой.

**Источники ошибок**

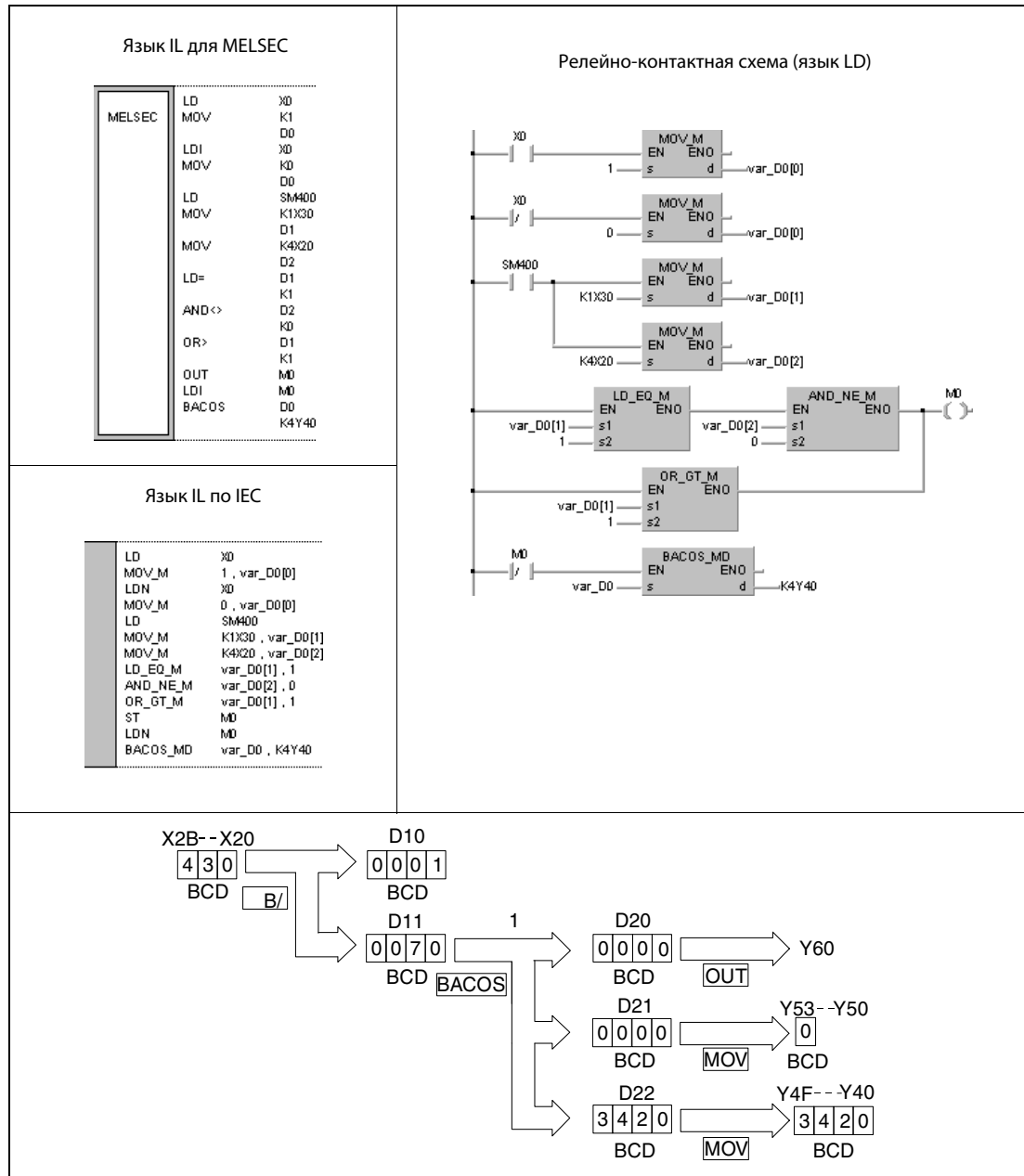
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Данные от  $s$  до  $s + 2$  не являются двоично-десятичными (код ошибки 4100).
- Данные, указанные от  $s$  до  $s + 2$ , находятся вне диапазона от  $-1.000$  до  $1.000$  (код ошибки 4100).

**Пример**

**BACOS**

Следующая программа при включенном маркере SM400 вычисляет арккосинус данных, имеющих следующий состав: арифметический знак на входе X0 (положительный, если X0 установлен; отрицательный, если X0 не установлен), 1-значная целочисленная часть в формате BCD на входах X30...X33 и 4-значная дробная часть в формате BCD на входах X20...X2F. Вычисленное значение угла выводится в виде 4-значного двоично-десятичного значения на выходы с Y40 по Y4F.



<sup>1</sup> вычисление арккосинуса

**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

7.12.19 BATAN, BATANP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

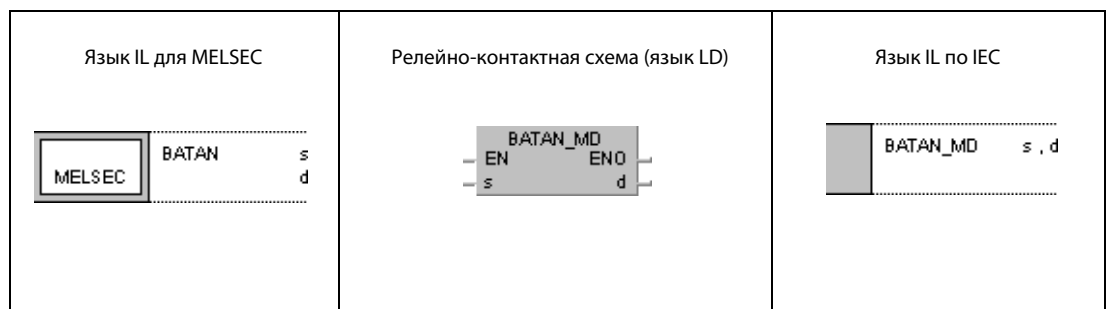
<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

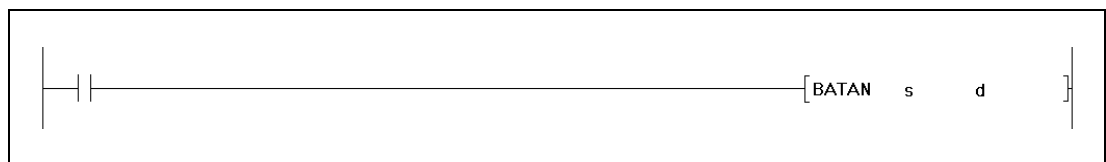
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	SM0	3	
d	●	●	●	●	●	●	—	—			

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
s	Первый адрес операнда со значением тангенса, в отношении которого выполняется команда BATAN (арктангенс).	BCD, 4 знака
d	Первый адрес операнда, в котором сохраняется результат.	

**Принцип действия**

**Вычисление арктангенса двоично-десятичных данных**

**BATAN Вычисление арктангенса**

Команда BATAN вычисляет угол по тангенсу, указанному в s, s + 1 и s + 2, и сохраняет результат в d.

$$\text{TAN}^{-1} = \left( \overset{s}{\boxed{\phantom{000}}} \overset{s+1}{\boxed{\phantom{000}}} . \overset{s+2}{\boxed{\phantom{000}}} \right) = d$$

1                      2                      3

- 1 арифметический знак
- 2 целочисленная часть
- 3 дробная часть

При положительном значении арифметический знак в s принимает значение 0, а при отрицательном значении – значение 1.

Целочисленная часть перед десятичной запятой и дробная часть после запятой могут быть только двоично-десятичными значениями в диапазоне между 0 и 9999.9999.

Результат в d может быть двоично-десятичным значением в диапазоне между 0° и 90° или между 270° и 360°.

Результат вычисления округляется, начиная с 5-го десятичного разряда после запятой.

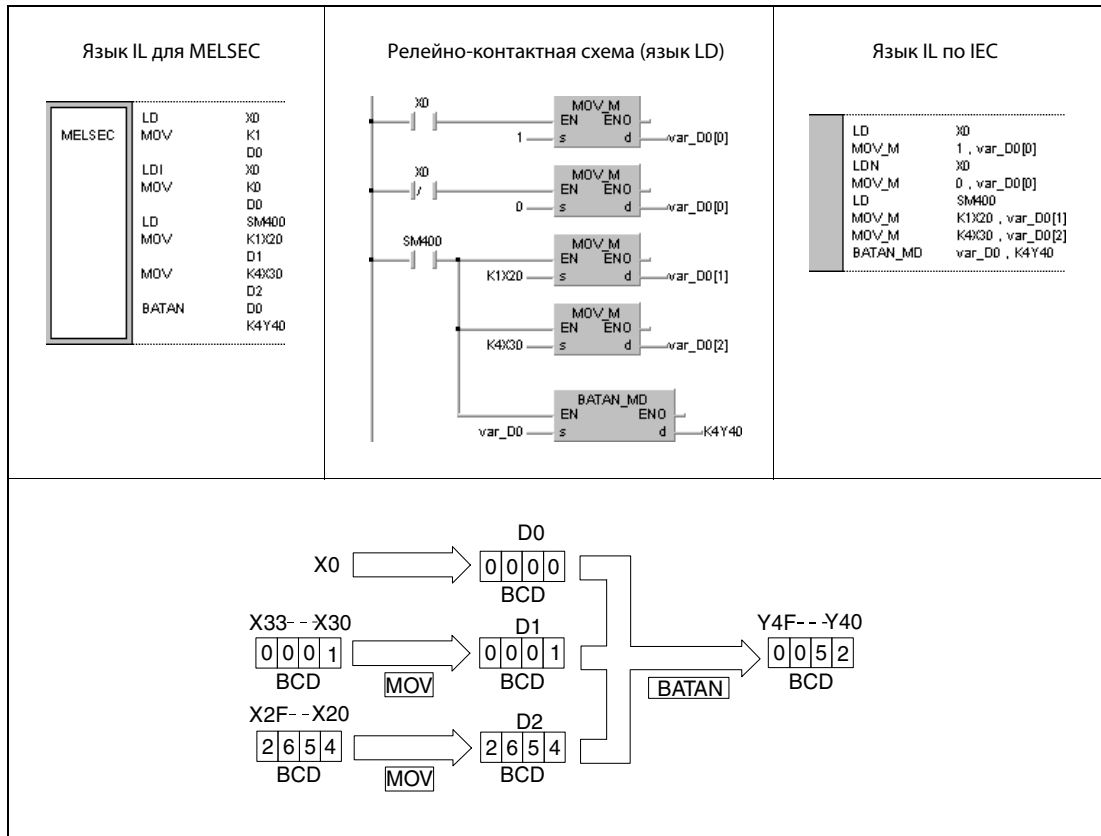
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Данные от s до s + 2 не являются двоично-десятичными (код ошибки 4100).

**Пример** BATAN

Следующая программа при включенном маркере SM400 вычисляет значение арктангенса из данных, имеющих следующий состав: арифметический знак на входе X0 (положительный, если X0 установлен; отрицательный, если X0 не установлен), 1-значная целочисленная часть в формате BCD на входах X20...X23 и 4-значная двоично-десятичная дробная часть на входах X30...X3F. Вычисленное значение угла выводится в виде 4-значного двоично-десятичного значения на выходы с Y40 по Y4F.



<sup>1</sup> вычисление арктангенса

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор или проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.



## 7.13 Команды контроля данных

Перечисленные в таблице команды контроля данных имеют входные и выходные операнды. После ограничения выходного значения или смещения входа/выхода 16-битные и 32-битные двоичные данные входных операндов выводятся на выходной операнд.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту МЭК (IEC)
Ограничение диапазона выходных значений 16-битных или 32-битных двоичных данных	LIMIT	LIMIT_MD
	LIMITP	LIMIT_P_MD
	DLIMIT	DLIMIT_MD
	DLIMITP	DLIMIT_P_MD
Смещение входных 16-битных или 32-битных двоичных данных	BAND	BAND_MD
	BANDP	BAND_P_MD
	DBAND	DBAND_MD
	DBANDP	DBAND_P_MD
Смещение выходных 16-битных или 32-битных двоичных данных	ZONE	ZONE_MD
	ZONEP	ZONE_P_MD
	DZONE	DZONE_MD
	DZONEP	DZONE_P_MD

**ПРИМЕЧАНИЕ** В редакторах по стандарту IEC используйте стандартные команды IEC.

## 7.13.1 LIMIT, LIMITP, DLIMIT, DLIMITP

## Процессор

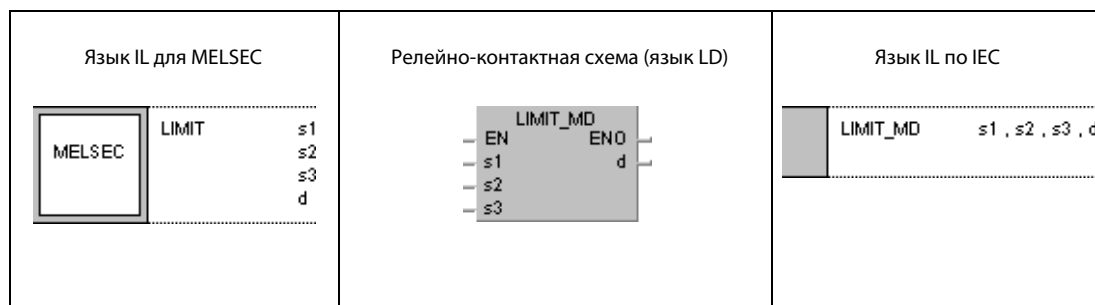
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	●

<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

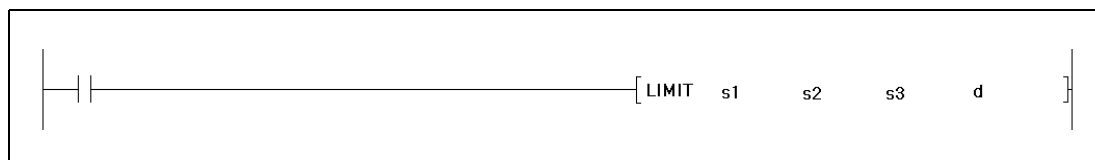
## Операнды MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s1	●	●	●	●	●	●	●	●	—	SM0	5
s2	●	●	●	●	●	●	●	●	—		
s3	●	●	●	●	●	●	●	●	—		
d	●	●	●	●	●	●	●	—	—		

## GX IEC Developer



## GX Developer



## Переменные

Операнд	Значение	Тип данных
s1	Нижний предел выходной величины (минимальное пороговое значение выхода)	BIN, 16 бит
s2	Верхний предел выходной величины (максимальное пороговое значение выхода)	
s3	Ограничиваемое входное значение	
d	Адрес операнда, в котором сохраняется ограниченное выходное значение.	

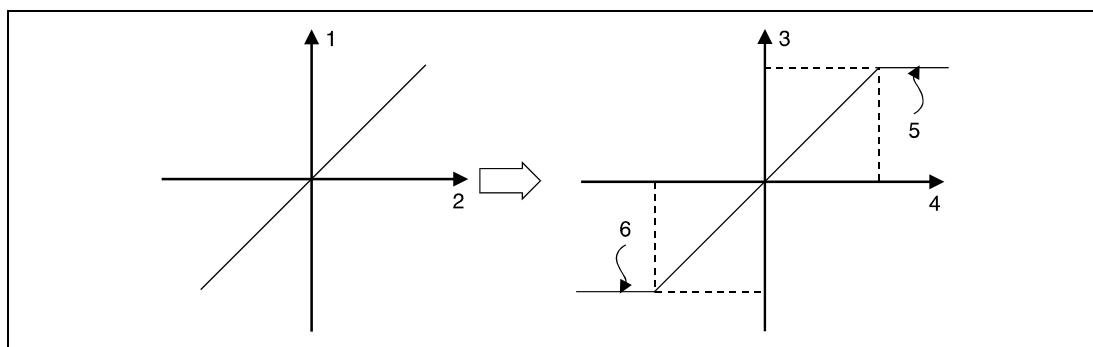
**Принцип работы****Ограничение диапазона выходных значений 16-битных и 32-битных двоичных данных****LIMIT Команда ограничения 16-битных двоичных данных**

Команда LIMIT проверяет данные, указанные в операнде s3 – находятся ли они между нижним пределом, указанным в s1, и верхним пределом, указанным в s2. В зависимости от результата проверки в операнде, указанном в d, сохраняется следующий результат:

Если указанное в s3 значение меньше нижнего предела, указанного в s1, то в операнде, указанном в d, сохраняется нижний предел.

Если указанное в s3 значение больше верхнего предела, указанного в s2, то в операнде, указанном в d, сохраняется верхний предел.

Если указанное в s3 значение находится между нижним и верхним пределом, то в операнде, указанном в d, сохраняется первоначальное значение.



- <sup>1</sup> выходное значение
- <sup>2</sup> входное значение
- <sup>3</sup> выходное значение (d)
- <sup>4</sup> входное значение (s3)
- <sup>5</sup> верхний предел (s2)
- <sup>6</sup> нижний предел (s1)

Значения, указанные в s1, s2 и s3, могут находиться в диапазоне от –32768 до 32767.

Если требуется контролировать только верхний предел, то для нижнего предела в s1 следует указать значение –32768.

Если требуется контролировать только нижний предел, то для верхнего предела в s2 следует указать значение 32767.

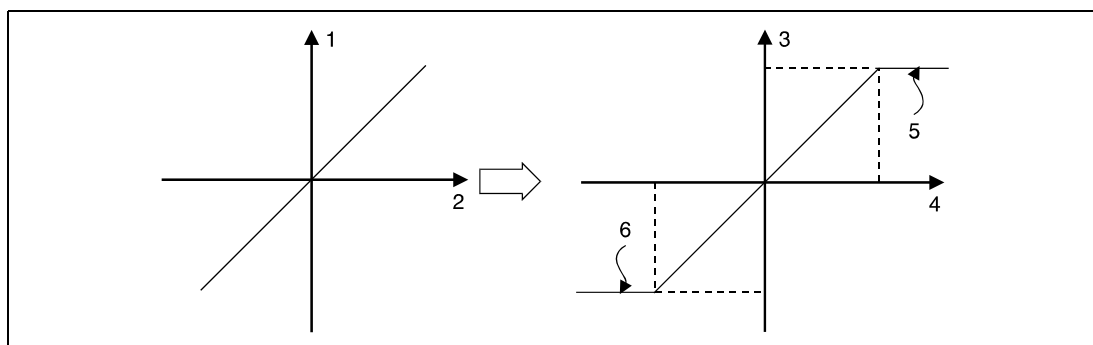
**DLIMIT Команда ограничения 32-битных двоичных данных**

Команда DLIMIT проверяет данные, указанные в операнде  $s3$  и  $(s3) + 1$  – находятся ли они между нижним пределом, указанным в  $s1$  и  $(s1) + 1$ , и верхним пределом, указанным в  $s2$  и  $(s2) + 1$ . В зависимости от результата проверки в операнде, указанном в  $d$  и  $d + 1$ , сохраняется следующий результат:

Если значение, указанное в  $s3$  и  $(s3) + 1$ , меньше нижнего предела, указанного в  $s1$  и  $(s1) + 1$ , то в операндах, указанных в  $d$  и  $d + 1$ , сохраняется нижний предел.

Если значение, указанное в  $s3$  и  $(s3) + 1$ , больше верхнего предела, указанного в  $s2$  и  $(s2) + 1$ , то в операндах, указанных в  $d$  и  $d + 1$ , сохраняется верхний предел.

Если значение, указанное в  $s3$  и  $(s3) + 1$ , находится между нижним и верхним пределом, то в операндах, указанных в  $d$  и  $d + 1$ , сохраняется первоначальное значение.



- <sup>1</sup> выходное значение
- <sup>2</sup> входное значение
- <sup>3</sup> выходное значение ( $d + 1, d$ )
- <sup>4</sup> входное значение ( $(s3) + 1, s3$ )
- <sup>5</sup> верхний предел ( $(s2) + 1, s2$ )
- <sup>6</sup> нижний предел ( $(s1) + 1, s1$ )

В  $s1$  и  $(s1) + 1$  и  $s2, (s2) + 1, s3$  и  $(s3) + 1$  можно указывать значения между  $-2147483648$  и  $2147483647$ .

Если требуется контролировать только верхний предел, то для нижнего предела в  $s1$  и  $(s1) + 1$  следует указать значение  $-2147483648$ .

Если требуется контролировать только нижний предел, то для верхнего предела в  $s2$  и  $(s2) + 1$  следует указать значение  $2147483647$ .

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Значение, указанное в  $s1$  ( $(s1) + 1$ ), превышает значение, указанное  $s2$  ( $(s2) + 1$ ) (код ошибки 4100).

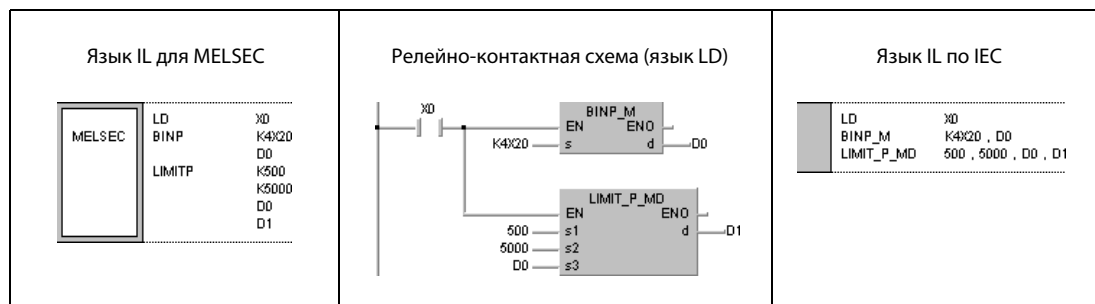
**Пример 1** LIMITP

В следующей программе при положительном фронте сигнала X0 проверяется, находится ли двоично-десятичное значение на входах с X20 по X2F между нижним пределом 500 и верхним пределом 5000. Результат проверки сохраняется в D1.

Если значение в D0 больше 5000, в D1 записывается 5000.

Если значение в D0 меньше 500, в D1 записывается 500.

Если значение в D0 находится между 500 и 5000, оно записывается в D1 без изменений.



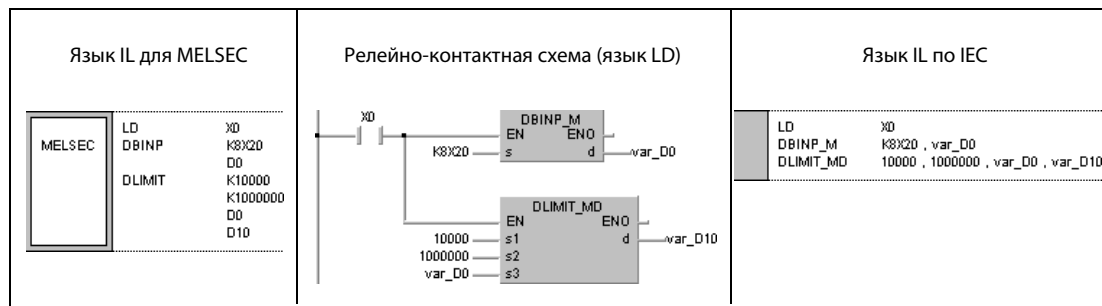
**Пример 2** DLIMIT

В следующей программе при положительном фронте сигнала X0 проверяется, находится ли двоично-десятичное значение на входах с X20 по X3F между нижним пределом 10000 и верхним пределом 1000000. Результат проверки сохраняется в D10 и D11.

Если значение в D0 и D1 больше 1000000, в D10 и D11 записывается 1000000.

Если значение в D0 и D1 меньше 10000, в D10 и D11 записывается 10000.

Если значение в D0 находится между 10000 и 1000000, оно записывается в D10 и D11 без изменений.



### 7.13.2 BAND, BANDP, DBAND, DBANDP

**Процессор**

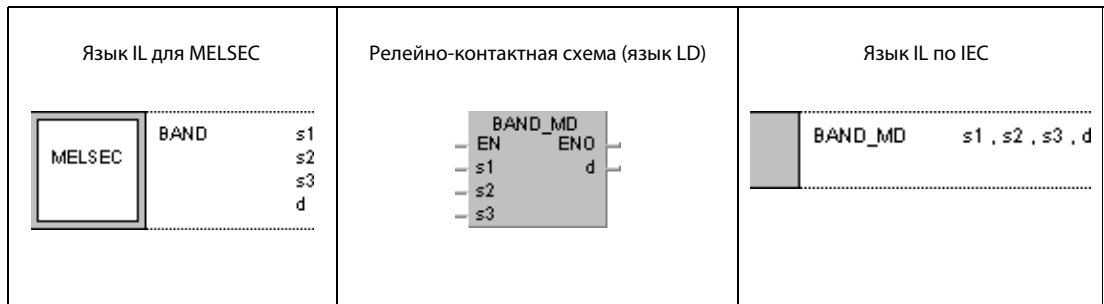
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	●

<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

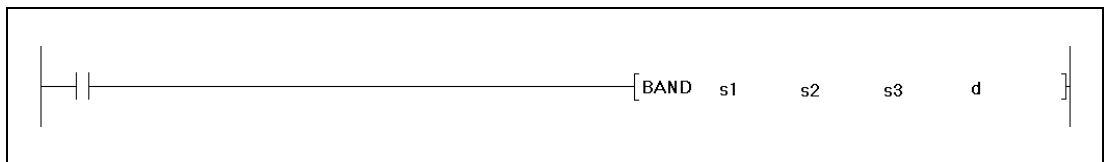
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	●	●	●	●	●	●	●	●	—	SM0	5
s2	●	●	●	●	●	●	●	●	—		
s3	●	●	●	●	●	●	●	●	—		
d	●	●	●	●	●	●	—	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s1	Нижнее смещение входа (начало диапазона, в котором выход = 0).	BIN, 16 бит
s2	Верхнее смещение входа (конец диапазона, в котором выход = 0).	
s3	Смещаемое входное значение	
d	Адрес операнда, в котором сохраняется разность входного значения и величины смещения.	

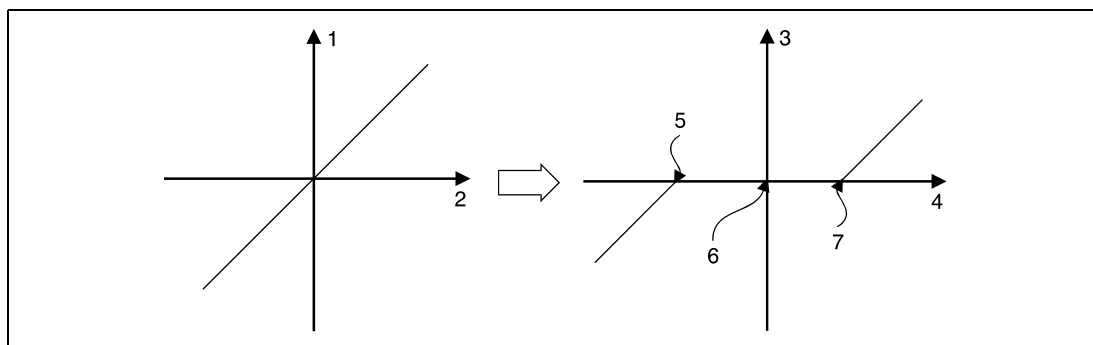
**Принцип работы****Смещение входных 16-битных и 32-битных двоичных данных****BAND Смещение входных 16-битных двоичных данных**

Команда BAND вычитает из указанного в s3 операнда указанное 16-битное двоичное значение нижнего смещения и верхнего смещения. Величина нижнего смещения сохранена в s1, а верхнего смещения – в s2. В зависимости от входного значения, результат сохраняется в операнде, указанном в d, следующим образом:

Если значение в s3 меньше величины нижнего смещения в s1, то в операнде, указанном в d, сохраняется разность  $s3 - s1$ .

Если значение в s3 больше величины верхнего смещения в s2, в операнде, указанном в d, сохраняется разность  $s3 - s2$ .

Если значение в s3 находится между вышеназванными величинами смещения, то в операнде, указанном в d, сохраняется 0.



1 выходное значение

2 входное значение

3 выходное значение (d)

4 входное значение (s3)

5 нижнее смещение входа (s1) (в данном примере отрицательное)

6 выходное значение = 0

7 верхнее смещение входа (s2) (в данном примере положительное)

Значения, указанные в s1, s2 и s3, могут находиться в диапазоне от -32768 до 32767.

Если разность выходит из диапазона -32768...32767, происходит следующее:

При занижении предела -32768 оставшаяся часть вычитания продолжается с числа 32767. Например, если в s3 сохранено значение -32768, и из него вычитается 10 (содержимое s1), то вычитание происходит следующим образом:

$$-32768 - 10 = 8000 \text{ н} - \text{An} = 7\text{FF}6\text{н} = 32758.$$

При превышении предела 32767 вычитание продолжается с числа -32768.

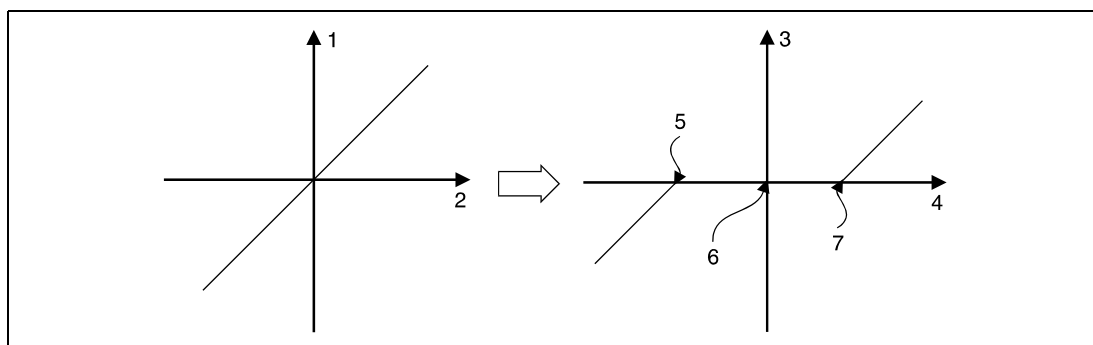
**DBAND Смещение входных 32-битных двоичных данных**

Команда DBAND вычитает из 32-битного двоичного числа, сохраненного в операндах  $s3$  и  $(s3) + 1$ , величину нижнего и верхнего смещения. Величина нижнего смещения сохранена в  $s1$  и  $(s1) + 1$ , а величина верхнего смещения – в  $s2$  и  $(s2) + 1$ . В зависимости от входного значения, результат сохраняется в указанных в  $d$  и  $d + 1$  операндах следующим образом:

Если значение в  $s3$  и  $(s3) + 1$  меньше величины нижнего смещения, указанного в  $s1$  и  $(s1) + 1$ , то в операнде, указанном в  $d$  и  $d + 1$ , сохраняется разность  $s3, (s3) + 1 - s1, (s1) + 1$ .

Если значение в  $s3$  и  $(s3) + 1$  больше величины верхнего смещения, указанного в  $s2$  и  $(s2) + 1$ , то в операнде, указанном в  $d$  и  $d + 1$ , сохраняется разность  $s3, (s3) + 1 - s2, (s2) + 1$ .

Если значение в  $s3$  и  $(s3) + 1$  находится между вышеназванными величинами смещения, то в операнд, указанный в  $d$  и  $d + 1$ , записывается 0.



<sup>1</sup> выходное значение

<sup>2</sup> входное значение

<sup>3</sup> выходное значение ( $d + 1, d$ )

<sup>4</sup> входное значение ( $(s3) + 1, s3$ )

<sup>5</sup> нижнее смещение входа ( $(s1) + 1, s1$ ) (в данном примере отрицательное)

<sup>6</sup> выходное значение = 0

<sup>7</sup> верхнее смещение входа ( $(s2) + 1, s2$ ) (в данном примере положительное)

Значения, указанные в  $s1$  и  $(s1) + 1$ ,  $s2$  и  $(s2) + 1$ ,  $s3$  и  $(s3) + 1$ , могут находиться в диапазоне между  $-2147483648$  и  $2147483647$ .

Если разность выходит из диапазона  $-2147483648$  и  $2147483647$ , происходит следующее:

При занижении предела  $-2147483648$  оставшаяся часть вычитания продолжается с числа  $2147483647$ . Например, если в  $s3$  и  $(s3) + 1$  указано значение  $-2147483648$ , и из него вычитается значение  $1000$ , указанное в  $s1$  и  $(s1) + 1$ , то вычитание происходит следующим образом:

$$-2147483648 - 1000 = 80000000_{\text{H}} - 3\text{E8}_{\text{H}} = 7\text{FFFFC18}_{\text{H}} = 2147482648.$$

При превышении предела  $2147483647$  вычитание продолжается с числа  $-2147483648$ .

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Значение, указанное в  $s1$  ( $(s1) + 1$ ), превышает значение, указанное  $s2$  ( $(s2) + 1$ ) (код ошибки 4100).



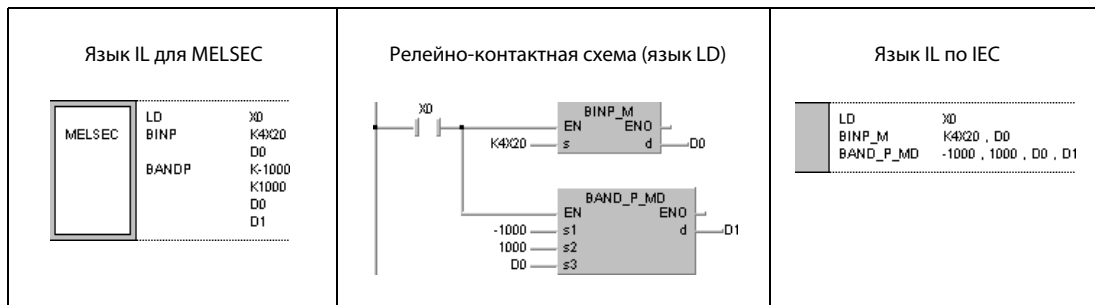
**Пример 1** BANDP

В следующей программе при положительном фронте сигнала X0 из двоично-десятичных данных на входах X20...X2F вычитается нижняя величина смещения -1000 и верхняя величина смещения 1000. Результат сохраняется в D1.

Если значение в D0 больше 1000, в D1 записывается разность  $D0 - 1000$ .

Если значение в D0 меньше -1000, в D1 записывается разность  $D0 - (-1000)$ .

Если значение в D0 находится между -1000 и 1000, в D1 записывается 0.



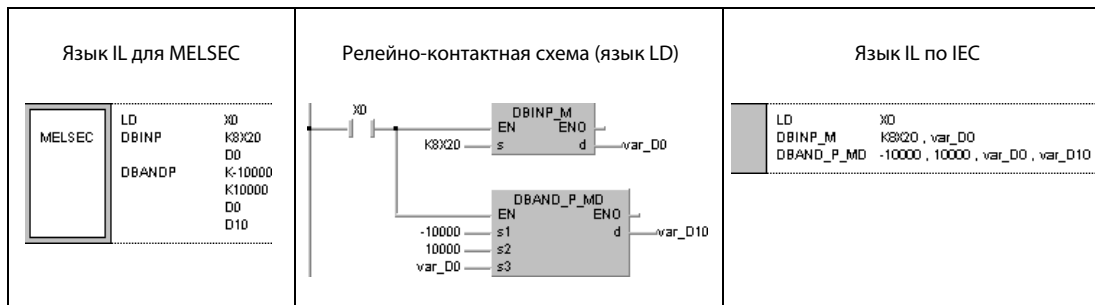
**Пример 2** DBANDP

В следующей программе при положительном фронте сигнала X0 из двоично-десятичных данных на входах X20...X3F вычитается нижняя величина смещения -10000 и верхняя величина смещения 10000. Результат сохраняется в D10 и D11.

Если значение в D0 и D1 больше 10000, в D10 и D11 записывается разность  $(D0, D1) - 10000$ .

Если значение в D0 и D1 меньше -10000, в D10 и D11 записывается разность  $(D0, D1) - (-10000)$ .

Если значение в D0 и D1 находится между -10000 и 10000, в D10 и D11 записывается 0.



### 7.13.3 ZONE, ZONEP, DZONE, DZONEP

**Процессор**

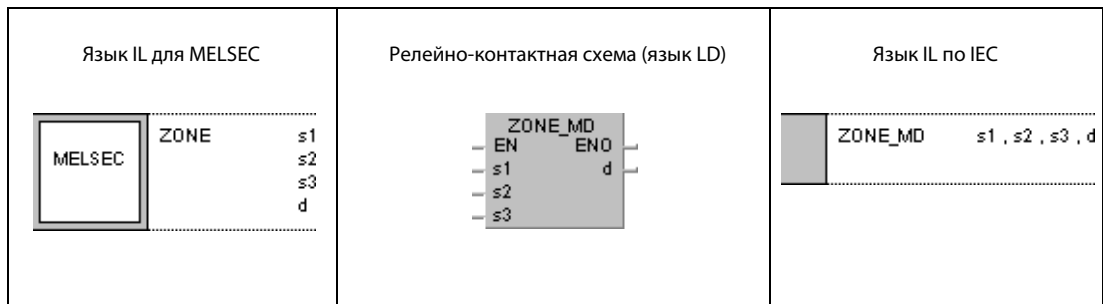
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	●

<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

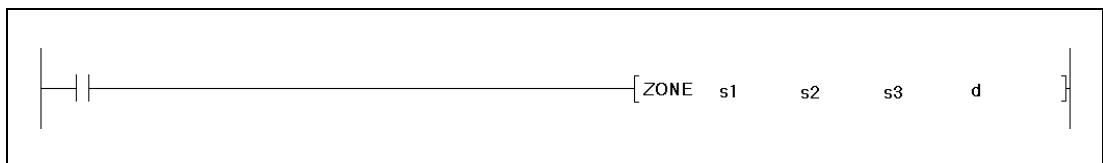
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непос. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s1	●	●	●	●	●	●	●	●	—	SM0	5
s2	●	●	●	●	●	●	●	●	—		
s3	●	●	●	●	●	●	●	●	—		
d	●	●	●	●	●	●	●	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s1	Нижняя величина смещения, прибавляемая к входному значению.	BIN, 16 бит
s2	Верхняя величина смещения, прибавляемая к входному значению.	
s3	Входное значение, в отношении которого выполняется команда смещения.	
d	Адрес операнда, в котором сохраняется сумма входного значения и величины смещения.	

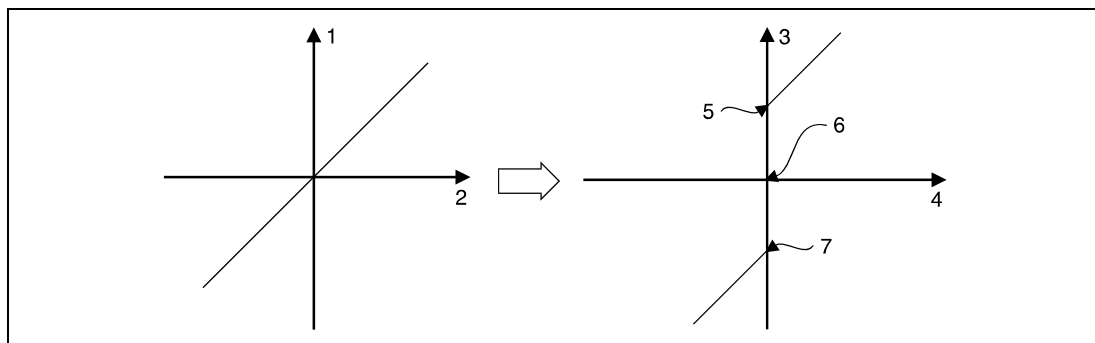
**Принцип действия****Смещение выходных 16-битных и 32-битных двоичных данных****ZONE Смещение выходных 16-битных двоичных данных**

Команда ZONE прибавляет к 16-битному двоичному числу в операнде, указанном в s3, нижнюю и верхнюю величину смещения. Нижняя величина смещения сохранена в s1, а верхняя – в s2. В зависимости от входного значения, результат сохраняется в операнде, указанном в d, следующим образом:

Если значение в s3 меньше 0, то в операнде, указанном в d, сохраняется сумма  $s3 + s1$ .

Если значение в s3 больше 0, то в операнде, указанном в d, сохраняется сумма  $s3 + s2$ .

Если значение данных в s3 равно 0, то в операнд, указанный в d, записывается 0.



1 выходное значение

2 входное значение

3 выходное значение (d)

4 входное значение (s3)

5 верхнее смещение выхода (s2) (в данном примере положительное)

6 входное значение = 0

7 нижнее смещение выхода (s1) (в данном примере отрицательное)

Значения, указанные в s1, s2 и s3, могут находиться в диапазоне от  $-32768$  до  $32767$ .

Если при сложении результат выходит из диапазона  $-32768...32767$ , происходит следующее:

При занижении предела  $-32768$  оставшаяся часть операции продолжается с числа  $32767$ . Например, если в s3 сохранено значение  $-32768$  и к нему прибавляется указанное в s1 значение  $-100$ , получается следующий результат:

$$-32768 + (-100) = 8000_{\text{H}} + \text{FF9C}_{\text{H}} = 7\text{F9C}_{\text{H}} = 32668.$$

При превышении предела  $32767$  оставшаяся часть сложения продолжается с числа  $-32768$ .

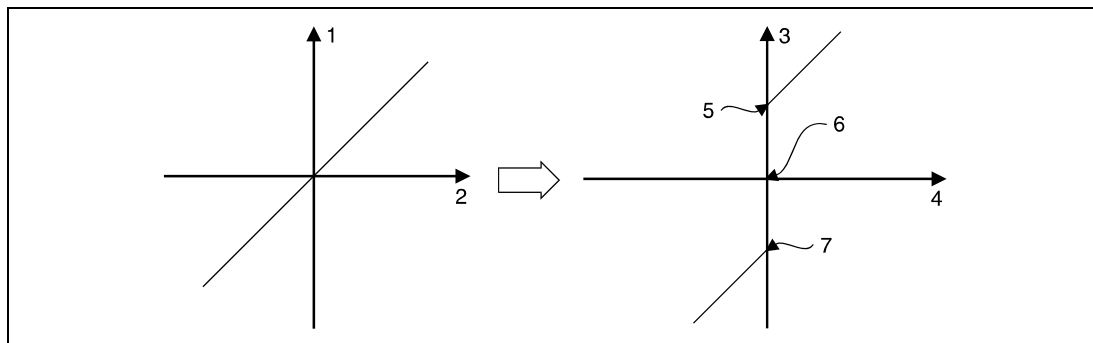
**DZONE Смещение выходных 32-битных двоичных данных**

Команда DZONE прибавляет к 32-битному двоичному числу в операнде, указанном в  $s3$  и  $(s3) + 1$ , нижнюю и верхнюю величину смещения. Нижняя величина смещения сохранена в  $s1$  и  $(s1) + 1$ , а верхняя – в  $s2$  и  $(s2) + 1$ . В зависимости от входного значения, результат сохраняется в указанных в  $d$  и  $d + 1$  операндах следующим образом:

Если значение в  $s3$  и  $(s3) + 1$  меньше 0, то в операнде, указанном в  $d$  и  $d + 1$ , сохраняется сумма  $(s3, (s3) + 1) + (s1, (s1) + 1)$ .

Если значение данных в  $s3$  и  $(s3) + 1$  больше 0, то в операнде, указанном в  $d$  и  $d + 1$ , сохраняется сумма  $(s3, (s3) + 1) + (s2, (s2) + 1)$ .

Если значение данных в  $s3$  и  $(s3) + 1$  равно 0, то в операнд, указанный в  $d$  и  $d + 1$ , записывается 0.



- <sup>1</sup> выходное значение
- <sup>2</sup> входное значение
- <sup>3</sup> выходное значение ( $d + 1, d$ )
- <sup>4</sup> входное значение  $((s3) + 1, s3)$
- <sup>5</sup> верхнее смещение выхода  $((s2) + 1, s2)$  (в данном примере положительное)
- <sup>6</sup> входное значение = 0
- <sup>7</sup> нижнее смещение выхода  $((s1) + 1, s1)$  (в данном примере отрицательное)

Значение, указанное в  $s1$  и  $(s1) + 1$ ,  $s2$  и  $(s2) + 1$ ,  $s3$  и  $(s3) + 1$ , может находиться в диапазоне от  $-2147483648$  до  $2147483647$ .

При занижении предела  $-2147483648$  оставшаяся часть операции продолжается с числа  $2147483647$ . Например, если в  $s3$  и  $(s3) + 1$  указано значение  $-2147483648$ , и к нему прибавляется значение  $-1000$ , указанное в  $s1$  и  $(s1) + 1$ , получается следующий результат:

$$-2147483648 + (-1000) = 80000000_{\text{H}} + \text{FFFFFFC18}_{\text{H}} = 7\text{FFFFFFC18}_{\text{H}} = 2147482648.$$

При превышении предела  $2147483647$  оставшаяся часть сложения продолжается с числа  $-2147483648$ .

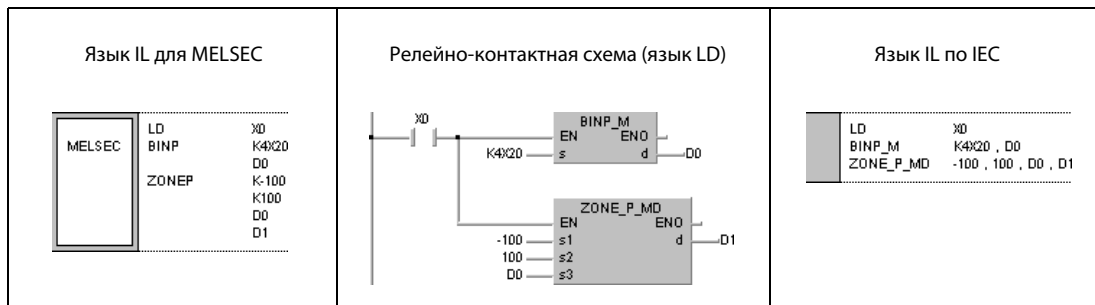
**Пример 1** ZONEP

В следующей программе при положительном фронте сигнала X0 к двоично-десятичным данным на входах X20...X2F прибавляется нижняя величина смещения -100 и верхняя величина смещения 100. Результат сохраняется в D1.

Если значение в D0 больше 0, в D1 записывается сумма  $D0 + 100$ .

Если значение в D0 меньше 0, в D1 записывается сумма  $D0 + (-100)$ .

Если значение в D0 равно 0, в D1 записывается 0.



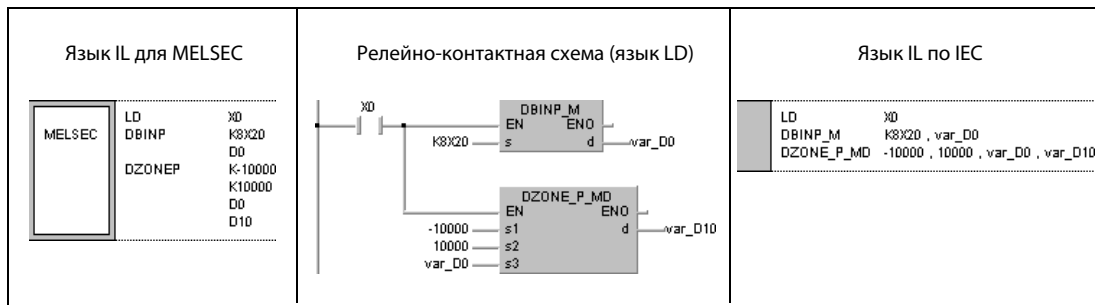
**Пример 2** DZONEP

В следующей программе при положительном фронте сигнала X0 к двоично-десятичным данным на входах X20...X3F прибавляется отрицательная величина смещения -10000 и положительная величина смещения 10 000. Результат сохраняется в D10 и D11.

Если значение в D0 и D1 больше 0, в D10 и D11 записывается сумма  $(D0, D1) + 10000$ .

Если значение в D0 и D1 меньше 0, в D10 и D11 записывается сумма  $(D0, D1) + (-10000)$ .

Если значение в D0 и D1 равно 0, в D10 и D11 записывается 0.



## 7.14 Команды для переключения блоков регистров файлов

Эти команды служат для переключения между блоками регистров файлов и между файлами регистров файлов. Обзор команд дан в таблице.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту МЭК (IEC)
Переключение между блоками регистров файлов	RSET	RSET_MD
		RSET_K_MD
	RSETP	RSET_P_MD
		RSET_K_P_MD
Переключение между файлами регистров файлов	QDRSET	QDRSET_M
	QDRSETP	QDRSET_P_MD
Переключение между файлами комментариев	QCDSSET	QCDSSET_M
	QCDSSETP	QCDSSET_P_MD

### 7.14.1 RSET, RSETP

**Процессор**

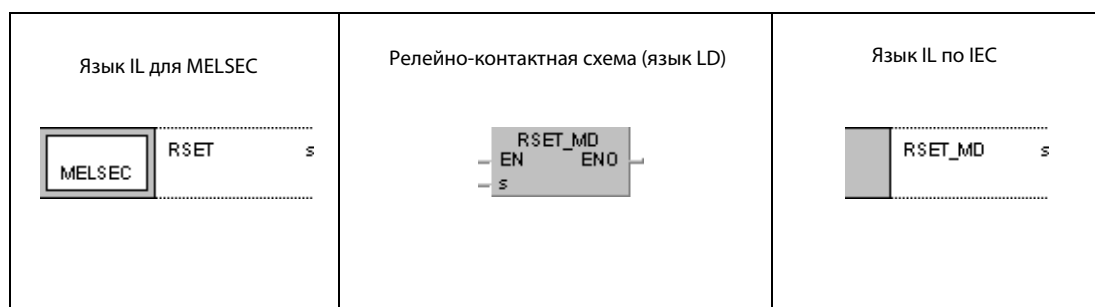
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	●

<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

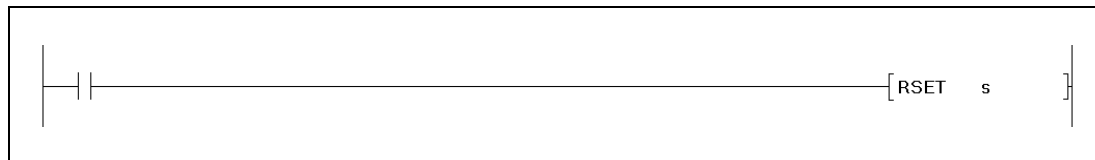
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	●	●	—	SМ0	2	

**GX IEC Developer**



**GX Developer**

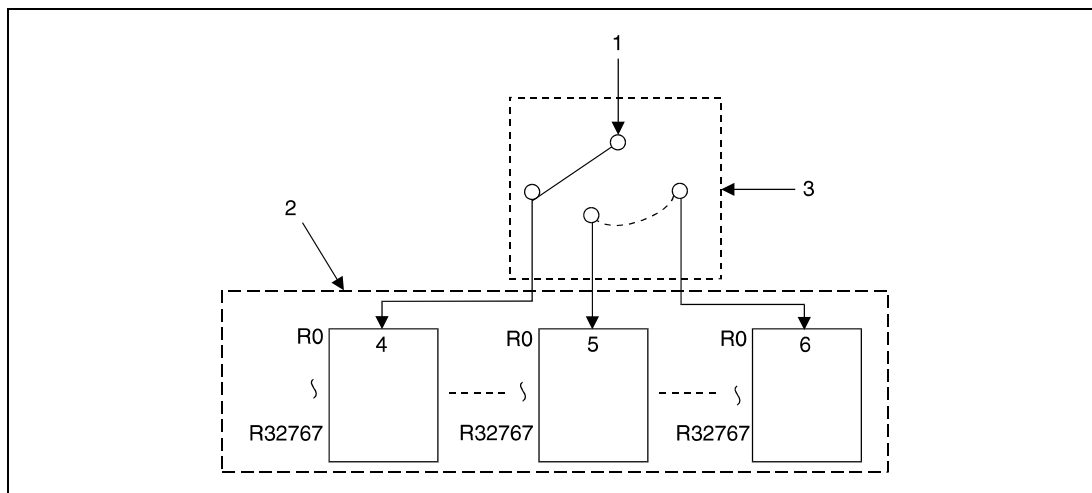


**Переменные**

Операнд	Значение	Тип данных
s	Адрес блока регистра файлов или операнд, в котором сохранен этот адрес.	BIN, 16 бит

**Принцип действия****Переключение между блоками регистров файлов****RSET Команда переключения блоков регистров файлов**

Команда RSET переключает с блока регистров файлов, используемого в программе, на блок регистров файлов, адрес которого указан в s. После переключения программа использует регистры файлов (R0–R32767) из блока, на который она была переключена.



<sup>1</sup> выполнение программы с обращением к регистрам файлов

<sup>2</sup> файл, используемый программой

<sup>3</sup> выбор блока регистров файлов (s)

<sup>4</sup> блок 0

<sup>5</sup> блок 1

<sup>6</sup> блок n

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

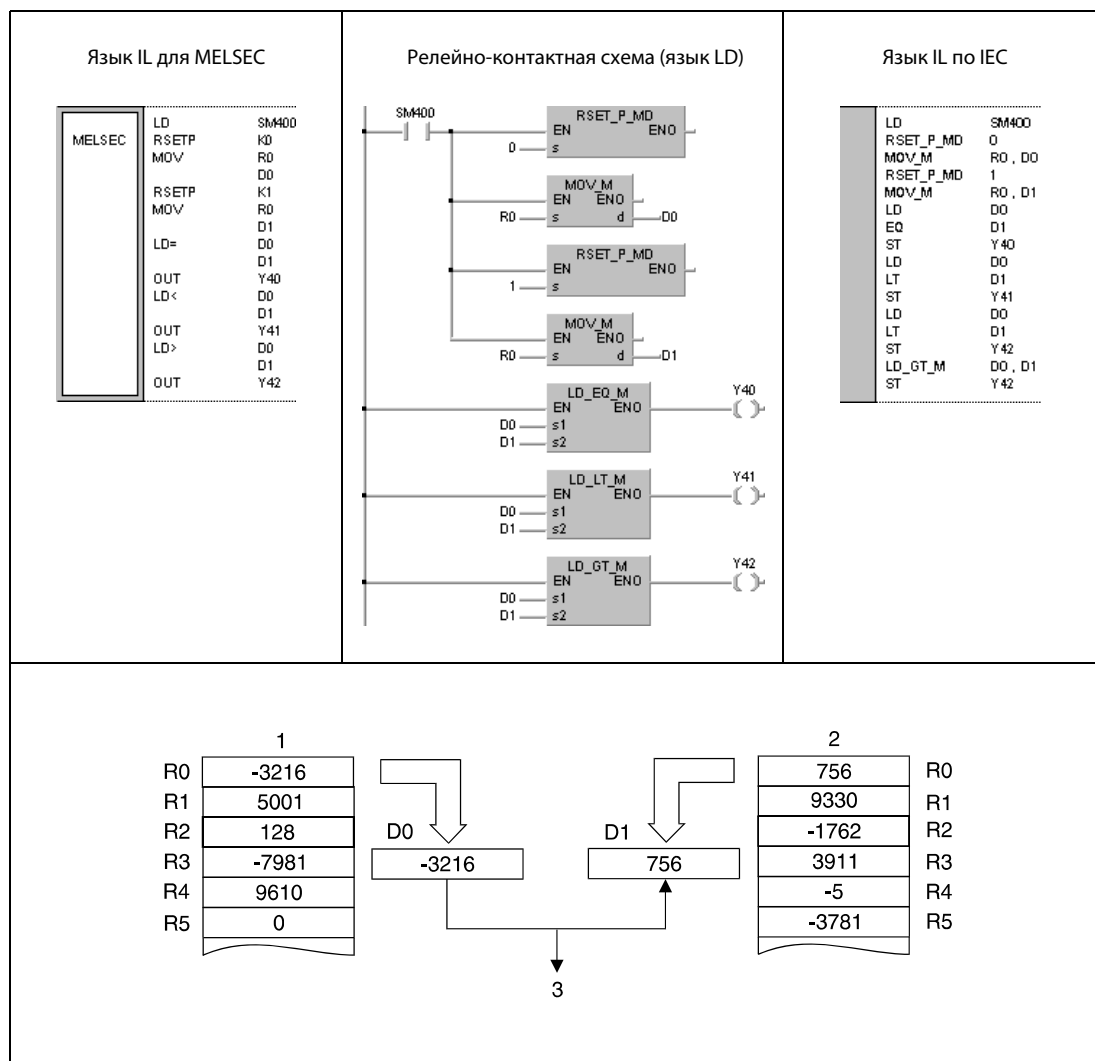
- По указанному в s адресу не имеется блока регистров файлов (код ошибки 4100).
- В блоке, указанном в s, нет регистров файлов (код ошибки 4101).



**Пример**

RSETP

В следующей программе при положительном фронте сигнала SM400 сравнивается регистр файлов R0 в блоках 0 и 1. Для адресации блоков 0 и 1 используется команда RSET. Считывание из обоих регистров R0 осуществляется с помощью команды MOV. Если значение в R0 блока 0 равно значению R0 блока 1, устанавливается выход Y40. Если значение R0 блока 0 меньше значения R0 блока 1, устанавливается выход Y41. Если значение R0 блока 0 больше значения R0 блока 1, устанавливается выход Y42.



<sup>1</sup> блок 0

<sup>2</sup> блок 1

<sup>3</sup> Y41 устанавливается, так как D0 меньше D1.

### 7.14.2 QDRSET, QDRSETP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

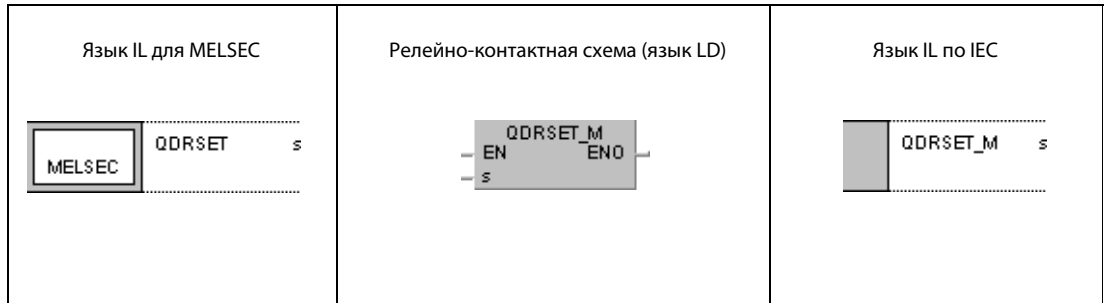
<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

**Операнды MELSEC Q**

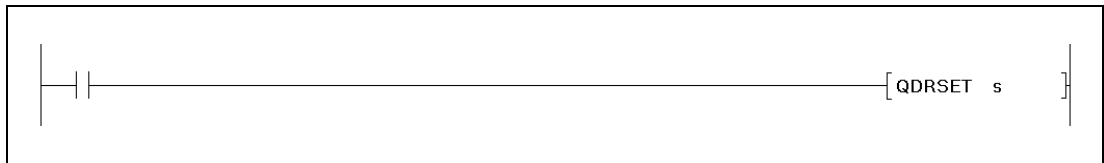
	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□\G□□	Индексные регистры Zn	Константы \$	Иные		
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	●	—	SM0	2 + n <sup>1)</sup>

<sup>1</sup> n = (количество знаков в названии файла/2) = количество дополнительных шагов (дробная часть округляется вверх).

**GX IEC Developer**



**GX Developer**

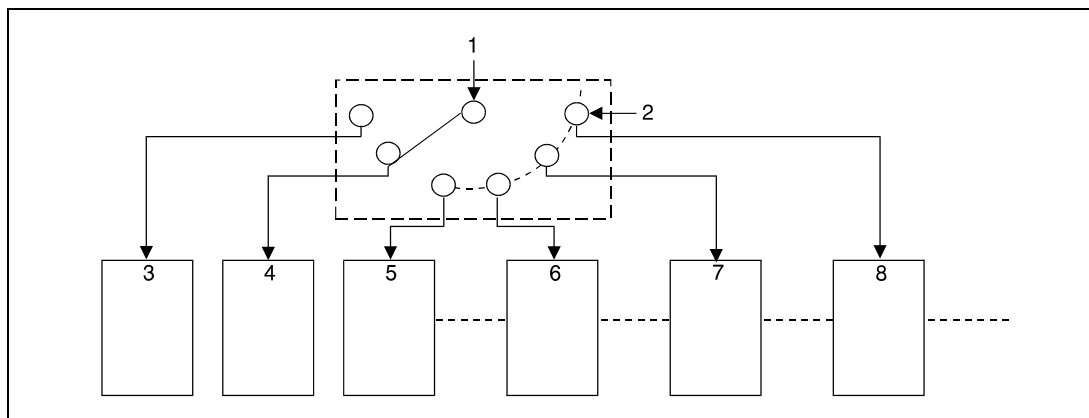


**Переменные**

Операнд	Значение	Тип данных
s	Адрес дисководов и название файла с регистрами файлов, на который переключается программа, или операнд, в котором сохранены эти данные.	строковая величина

**Принцип действия****Переключение между файлами регистров файлов****QDRSET****Команда для переключения файлов регистров файлов**

Команда QDRSET переключает программу с используемого файла регистров файлов на файл регистров файлов, указанный в *s*. После переключения программа использует регистры файлов (R0–R32767) в блоке 0 файла, на который было осуществлено переключение. Для переключения между блоками используется команда RSET.



<sup>1</sup> выполнение программы с обращением к регистрам файлов

<sup>2</sup> выбор дисководов и файла (*s*)

<sup>3</sup> дисковод 1, файл А

<sup>4</sup> дисковод 1, файл В

<sup>5</sup> дисковод 1, файл С

<sup>6</sup> дисковод 2, файл А

<sup>7</sup> дисковод 3, файл А

<sup>8</sup> дисковод 4, файл А

Возможна адресация 4 дисководов (1–4). Дисководу не может быть присвоен адрес 0, так как он зарезервирован для внутренней памяти.

Указывать расширение. QDR при обозначении файла регистров файлов не требуется.

Для сброса выбранного названия файла следует в качестве названия файла указать 0 в шестнадцатеричной кодировке (00h).

Файлы регистров файлов, на которые программа была переключена с помощью команды QDRSET, имеют приоритет обработки даже в том случае, если адрес дисковода и название файла регистров файлов были указаны в параметрах.

**Источники ошибок**

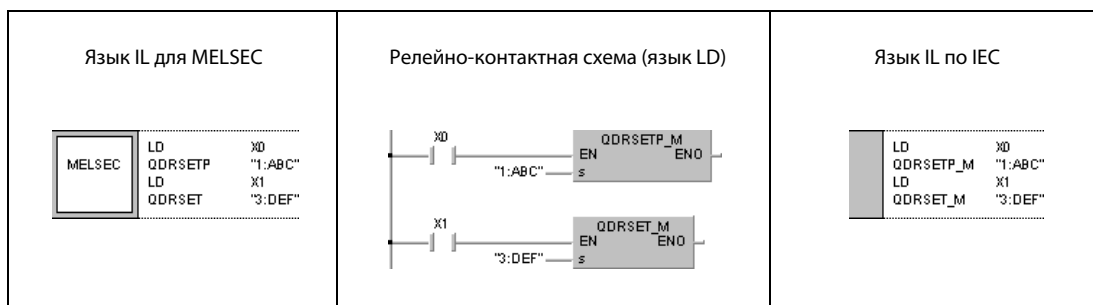
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В дисковом, указанном в *s*, нет файла регистров файлов (код ошибки 2410).

**Пример**

## QDRSET/QDRSETP

В следующей программе при положительном фронте сигнала X0 происходит переключение на файл регистров файлов ABC.QDR в дисководе 1. При включенном входе X1 происходит переключение на файл регистров файлов DEF.QDR в дисководе 3.



### 7.14.3 QCDSET, QCDSETP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

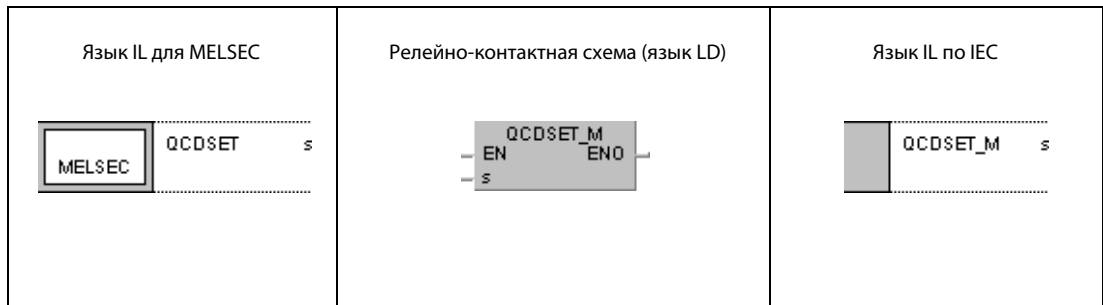
<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

**Операнды MELSEC Q**

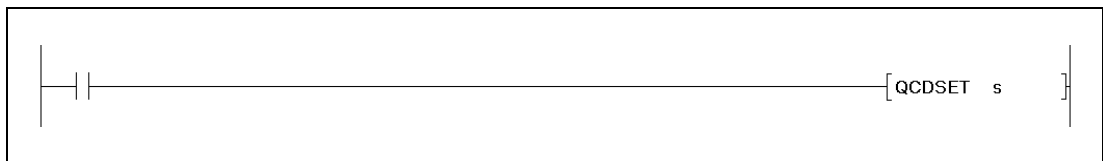
	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непос. адр. J□□□		Спец. модули U□\G□□	Индексные регистры Zn	Константы \$	Иные		
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	●	—	SM0	2+n <sup>1)</sup>

<sup>1</sup> n = (количество знаков в названии файла/2) = количество дополнительных шагов (дробная часть округляется вверх).

**GX IEC Developer**



**GX Developer**

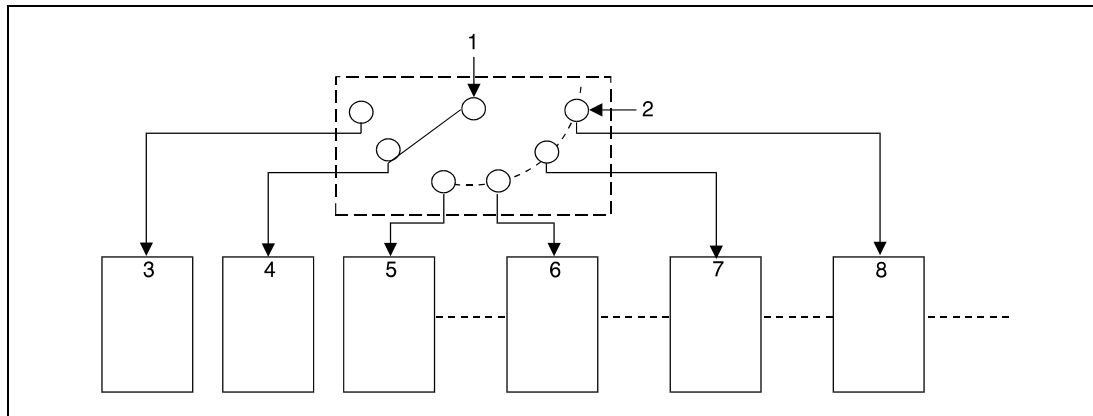


**Переменные**

Операнд	Значение	Тип данных
s	Адрес дисковогода и название файла комментариев, на который осуществляется переключение, или операнд, в котором сохранены эти данные.	строковая величина

**Принцип действия****Переключение между файлами комментариев****QCDSET****Команда переключения файлов комментариев**

Команда QCDSET переключает программу с используемого в данный момент файла комментариев на файл, указанный в *s*. После переключения программа использует данные комментариев из файла, на который она была переключена.



1 программа с обращением к данным комментариев

2 выбор дисководов и файла комментариев (*s*)

3 дисковод 1, файл комментариев "А"

4 дисковод 1, файл комментариев "В"

5 дисковод 1, файл комментариев "С"

6 дисковод 2, файл комментариев "А"

7 дисковод 3, файл комментариев "А"

8 дисковод 4, файл комментариев "А"

Возможна адресация 4 дисководов (1–4). Дисководу не может быть присвоен адрес 0, так как он зарезервирован для внутренней памяти.

При указании файлов комментариев расширение .QCD указывать не требуется.

Для сброса выбора файла следует в качестве названия файла ввести нулевой шестнадцатеричный код (00h).

Файлы комментариев, на которые программа была переключена с помощью команды QCDSET, имеют приоритет обработки даже в том случае, если адреса дисководов и названия файлов комментариев были указаны в параметрах.

**Источники ошибок**

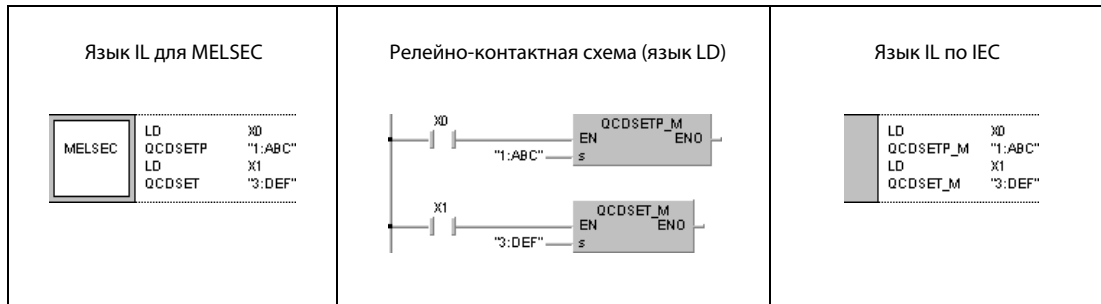
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В указанном в *s* дисководе нет файла комментариев (код ошибки 2410).

**Пример**

QCDSET/QCDSETP

В следующей программе при положительном фронте сигнала X0 происходит переключение на файл комментариев ABC.QCD в дисковом 1. При включенном входе X1 происходит переключение на файл комментариев DEF.QCD в дисковом 3.



## 7.15 Команды для часов

Эти команды позволяют считывать и записывать данные времени, выполнять вычислительные операции с данными времени и изменять формат данных времени. Обзор команд содержится в таблице.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту МЭК (IEC)
Считывание данных времени	DATERD	DATERD_MD
	DATERDP	DATERD_P_MD
Запись данных времени	DATEWR	DATEWR_MD
	DATEWRP	DATEWR_P_MD
Сложение данных времени	DATE+	DATEPLUS_M
	DATE+P	DATEPLUSP_M
Вычитание данных времени	DATE-	DATEMINUS_M
	DATE-P	DATEMINUSP_M
Смена формата времени с "часов, минут и секунд" на секунды	SECOND	SECOND_M
	SECONDP	SECONDP_M
Смена формата времени с секунд на "часы, минуты и секунды"	HOUR	HOUR_M
	HOURP	HOURP_M



### 7.15.1 DATERD, DATERDP

**Процессор**

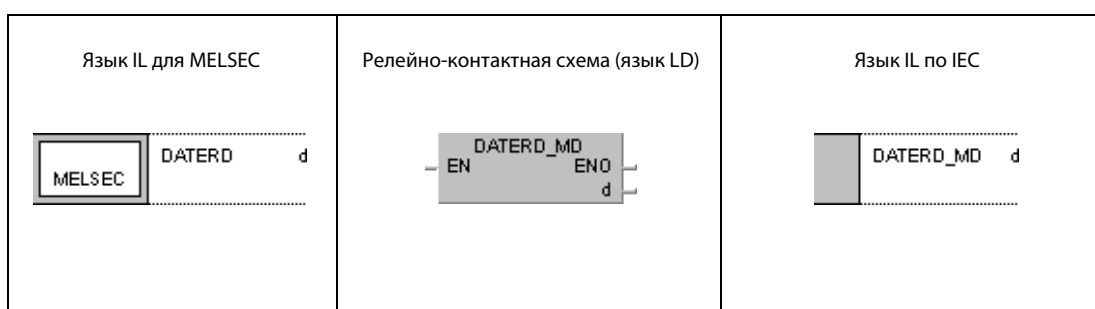
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	●

<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

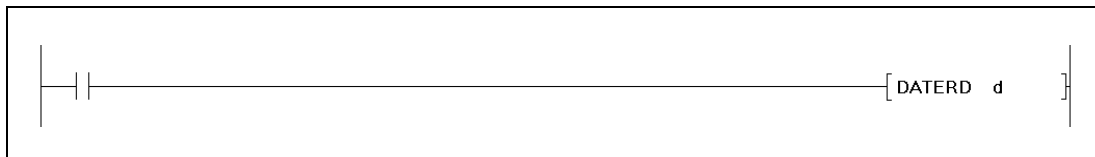
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непос. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы			Иные
	битовые	словные		битовые	словные						
d	—	●	●	—	—	—	—	—	—	2	

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных	
		MELSEC	IEC
d	Первый адрес операнда, в котором сохраняются считанные данные времени.	BIN, 16 бит	массив [0..6] данных типа ANY16

**Принцип действия**

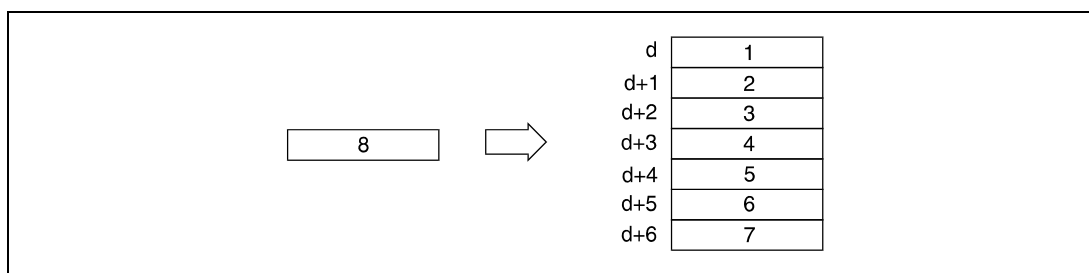
**Считывание данных времени**

**DATERD Команда считывания**

Команда DATERD считывает год, месяц, день, час, минуту, секунду и день недели из внутренних часов ЦПУ и сохраняет эти данные в двоичном формате в операнде, указанном с  $d + 0$  (Array\_d[1]) по  $d + 6$  (Array\_d[7]). Взаимосвязь между регистрами и данными времени пояснена на следующей иллюстрации:

- $d + 0$ , Array\_d[1] = год (1)
- $d + 1$ , Array\_d[2] = месяц (январь = 1, декабрь = 12) (2)
- $d + 2$ , Array\_d[3] = день (3)
- $d + 3$ , Array\_d[4] = час (24-часовой формат) (4)
- $d + 4$ , Array\_d[5] = минута (5)
- $d + 5$ , Array\_d[6] = секунда (6)
- $d + 6$ , Array\_d[7] = день недели (7)

Часы центрального процессора на иллюстрации обозначены цифрой 8.



В следующей таблице указаны диапазоны значений, сохраненных в операндах с  $d + 0$  (Array\_d[1]) по  $d + 6$  (Array\_d[7]).

Данные времени	Год	Месяц	День	Час	Минута	Секунда	День недели
Диапазон	1)	1–12	1–31	0–23	0–59	0–59	0–6
Операнд	$d + 0$ (Array_d[0])	$d + 1$ (Array_d[1])	$d + 2$ (Array_d[2])	$d + 3$ (Array_d[3])	$d + 4$ (Array_d[4])	$d + 5$ (Array_d[5])	$d + 6$ (Array_d[6])

<sup>1</sup> Диапазон в случае процессоров QnA: от 0 до 99, в случае процессоров System Q: от 1980 до 2079.

В процессорах QnA год сохраняется только в виде двух цифр: разряда единиц и разряда десятков (например, 1995 = 95).

В процессоре "System Q" год сохраняется в виде четырехзначного числа.

День недели в  $D + 6$  (Array\_d[7]) указывается в диапазоне от 0 до 6. Обозначение дней недели разъяснено в следующей таблице.

День недели	Воскресенье	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота
Сохраняемые значения	0	1	2	3	4	5	6

Часы центрального процессора автоматически определяют високосные года.

**Пример 1** DATERD (процессоры QnA)

В следующей программе при включенном маркере SM400 данные времени считываются из внутренних часов ЦПУ и выводятся на выходы Y47–Y67 в виде двоично-десятичных данных следующим образом.

Y60 – Y67 = месяц

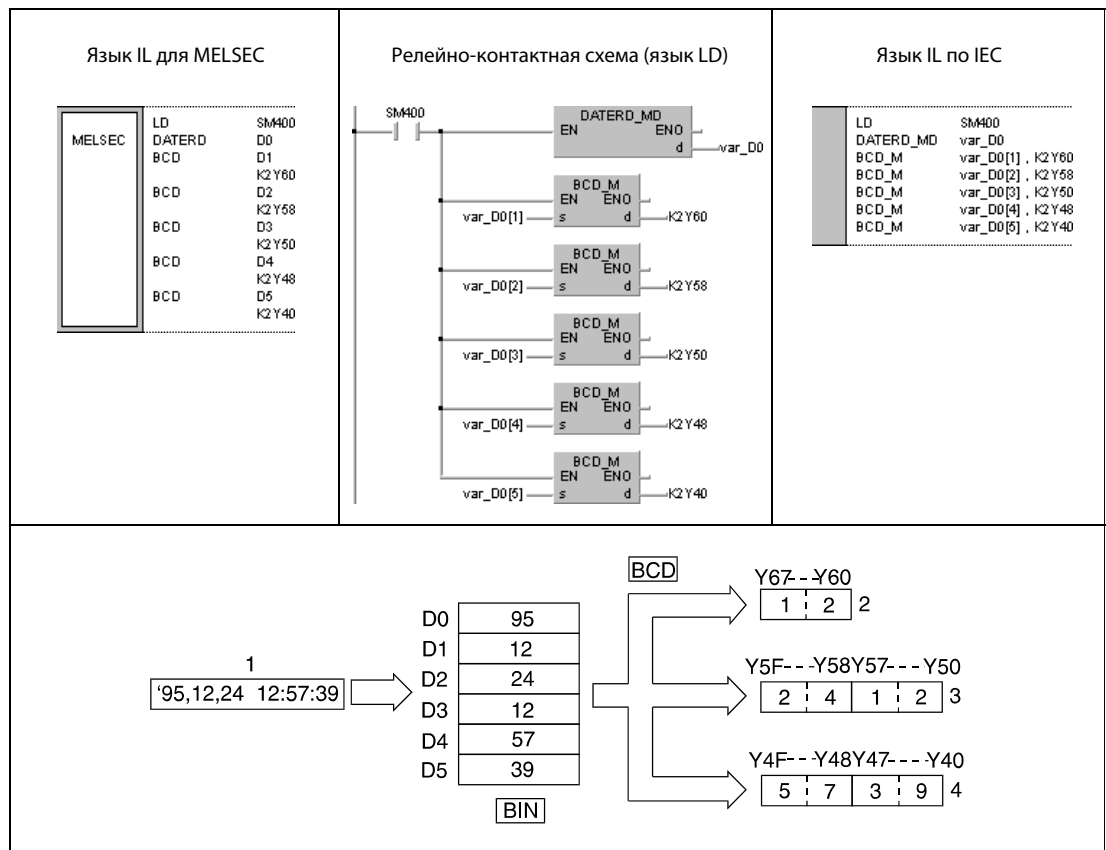
Y58 – Y5F = день

Y50 – Y57 = час

Y48 – Y4F = минута

Y40 – Y47 = секунда

Данные времени	Год	Месяц	День	Час	Минута	Секунда	День недели
Операнд	D0 (var_D0[0])	D1 (var_D0[1])	D2 (var_D0[2])	D3 (var_D0[3])	D4 (var_D0[4])	D5 (var_D0[5])	D6 (var_D0[6])



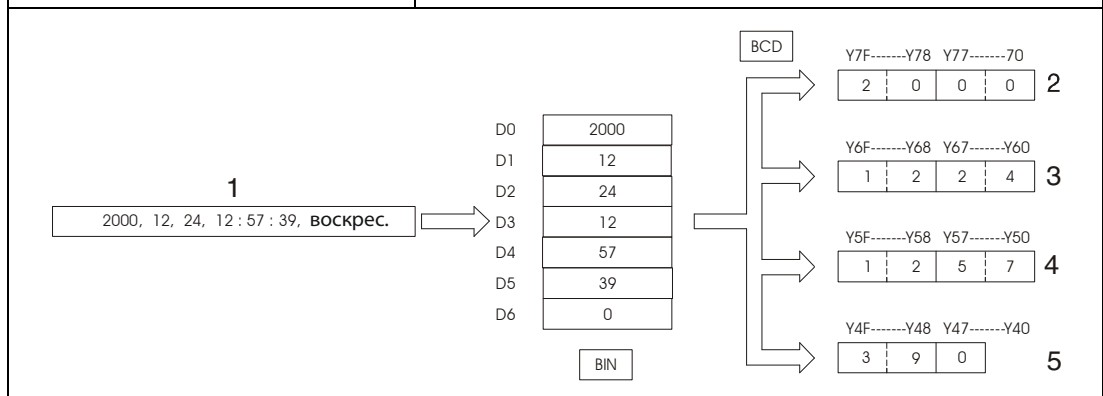
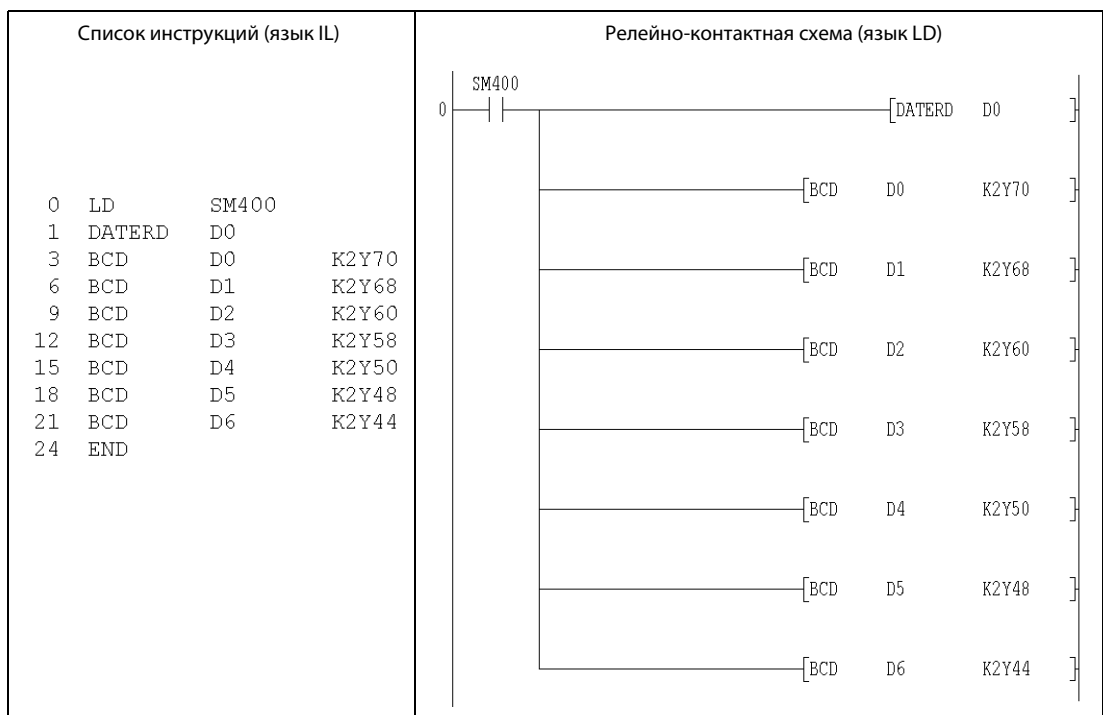
**ПРИМЕЧАНИЕ** *Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.*

**Пример 2** DATE RD (процессор серии "System Q")

В следующей программе при включенном маркере SM400 данные времени считываются из внутренних часов ЦПУ и выводятся на выходы Y47–Y67 в виде двоично-десятичных данных следующим образом.

- Y70 – Y7F = год
- Y68 – Y6F = месяц
- Y60 – Y67 = день
- Y58 – Y5F = час
- Y50 – Y57 = минута
- Y48 – Y4F = секунда
- Y44 – Y47 = день недели

Данные времени	Год	Месяц	День	Час	Минута	Секунда	День недели
Операнд	D0	D1	D2	D3	D4	D5	D6



- <sup>1</sup> данные времени
- <sup>2</sup> год
- <sup>3</sup> месяц, день
- <sup>4</sup> час, минута,
- <sup>5</sup> секунда, день недели

## 7.15.2 DATEWR, DATEWRP

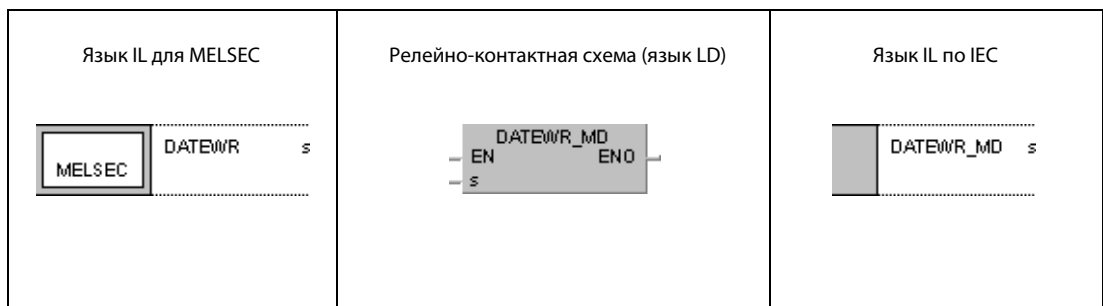
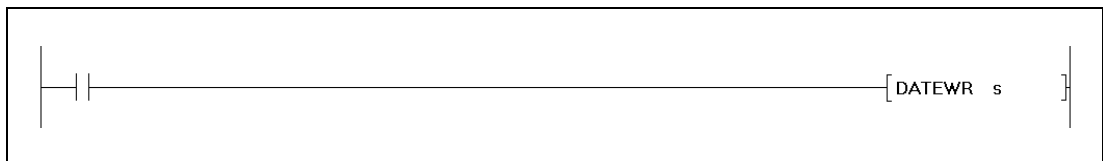
## Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	●

<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)			Иные DY
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	—	SM0	2

GX IEC  
DeveloperGX  
Developer

## Переменные

Операнд	Значение	Тип данных	
		MELSEC	IEC
s	Первый адрес операнда, в котором сохранены данные, записываемые во внутренние часы ЦПУ.	BIN, 16 бит	массив [0..6] данных типа ANY16

**Принцип действия****Запись данных времени****DATEWR Команда записи**

Команда DATEWR записывает данные времени (год, месяц, день, час, минуту, секунду и день недели), которые хранятся в операндах, указанных с  $s + 0$  (Array\_s[1]) по  $s + 6$  (Array\_s[7]), во внутренние часы ЦПУ. Данные времени хранятся в двоичном формате. Взаимосвязь между регистрами и данными времени, записываемыми во внутренние часы ЦПУ, пояснена на следующей иллюстрации:

$s + 0$ , Array\_s[1] = год (1)

$s + 1$ , Array\_s[2] = месяц (январь = 1, декабрь = 12) (2)

$s + 2$ , Array\_s[3] = день (3)

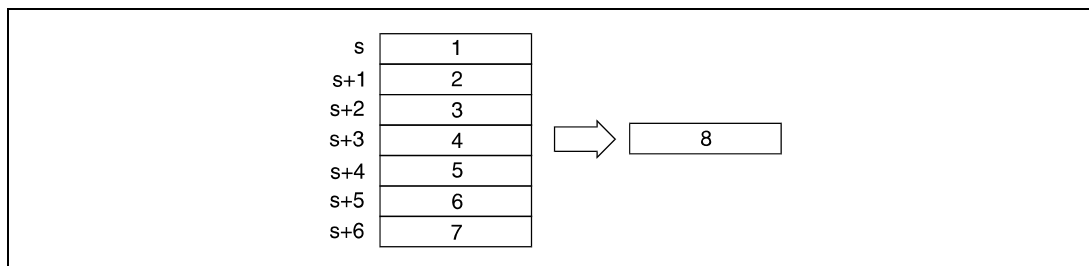
$s + 3$ , Array\_s[4] = час (24-часовой формат, от 0 до 23 часов) (4)

$s + 4$ , Array\_s[5] = минута (5)

$s + 5$ , Array\_s[6] = секунда (6)

$s + 6$ , Array\_s[7] = день недели (7)

Часы центрального процессора на иллюстрации обозначены цифрой 8.



В следующей таблице указаны диапазоны значений, сохраненных в операндах с  $s + 0$  (Array\_s[1]) по  $s + 6$  (Array\_s[7]).

Данные времени	Год	Месяц	День	Час	Минута	Секунда	День недели
Диапазон	1)	1–12	1–31	0–23	0–59	0–59	0–6
Операнд	$s + 0$ (Array_s[0])	$s + 1$ (Array_s[1])	$s + 2$ (Array_s[2])	$s + 3$ (Array_s[3])	$s + 4$ (Array_s[4])	$s + 5$ (Array_s[5])	$s + 6$ (Array_s[6])

<sup>1</sup> Диапазон в случае процессоров QnA: от 0 до 99; в случае процессоров "System Q": от 1980 до 2079

В процессорах QnA год сохраняется только в виде двух цифр – разряда единиц и разряда десятков (например, 1995 = 95). В процессоре "System Q" сохраняется четырехзначное число года.

День недели, сохраненный в  $s + 6$  (Array\_s[7]), указывается в диапазоне от 0 до 6. Обозначение дней недели разъяснено в следующей таблице.

День недели	Воскресенье	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота
Сохраняемые значения	0	1	2	3	4	5	6

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

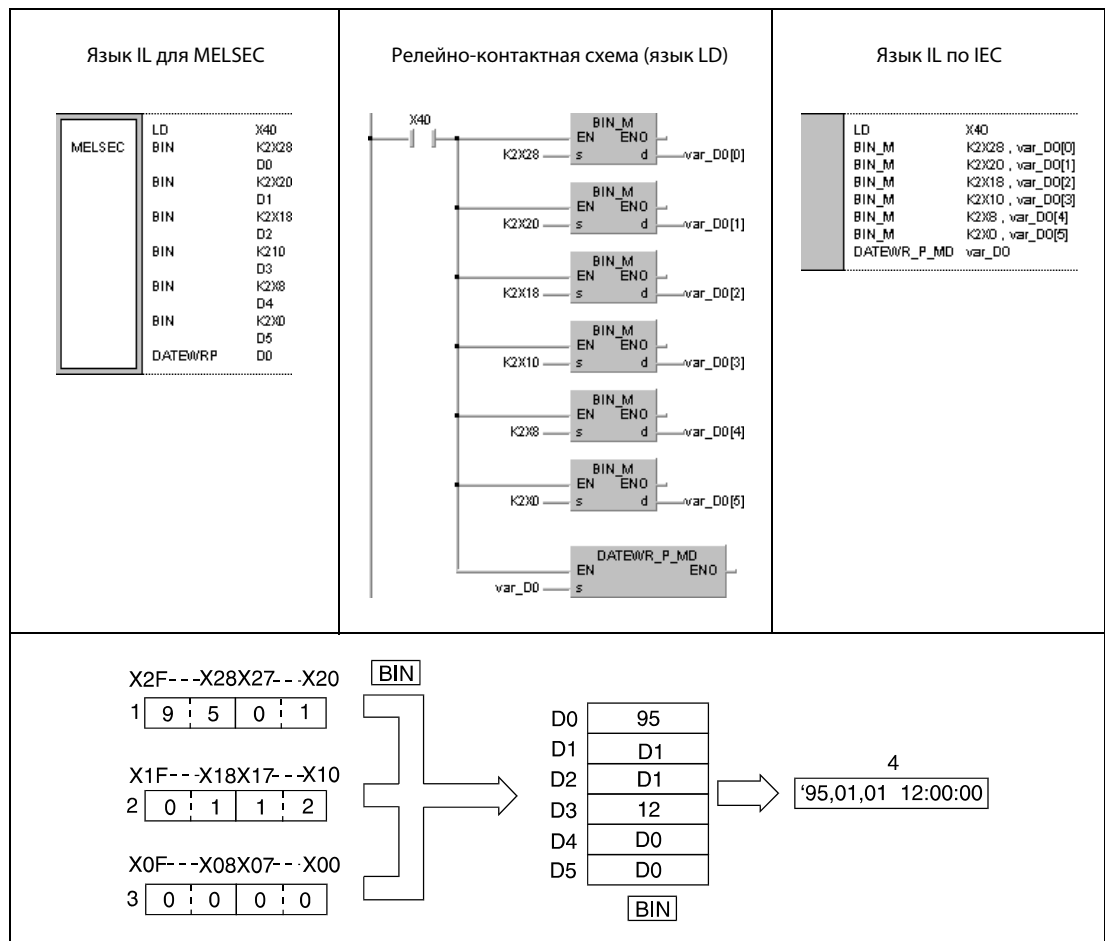
- Данные времени, указанные в операндах с  $s + 0$  (Array\_s[1]) по  $s + 6$  (Array\_s[7]), находятся вне диапазона (код ошибки 4100).

**Пример 1** DATEWRP (процессор QnA)

В следующей программе при положительном фронте сигнала X40 данные времени, поданные на входы X0...X2F в формате BCD, записываются во внутренние часы ЦПУ. Входы сопоставлены данным времени следующим образом.

- X28 – X2F = год
- X20 – X27 = месяц
- X18 – X1F = день
- X10 – X17 = час
- X8 – XF = минута
- X0 – X7 = секунда

Данные времени	Год	Месяц	День	Час	Минута	Секунда	День недели
Операнд	D0 (var_D0[0])	D1 (var_D0[1])	D2 (var_D0[2])	D3 (var_D0[3])	D4 (var_D0[4])	D5 (var_D0[5])	D6 (var_D0[6])



- 1 год, месяц
- 2 день, час
- 3 минута, секунда
- 4 данные времени



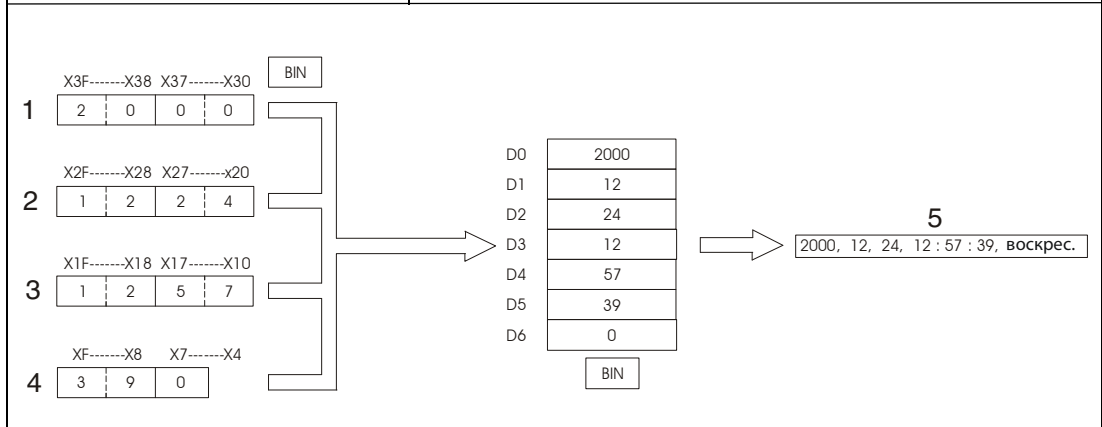
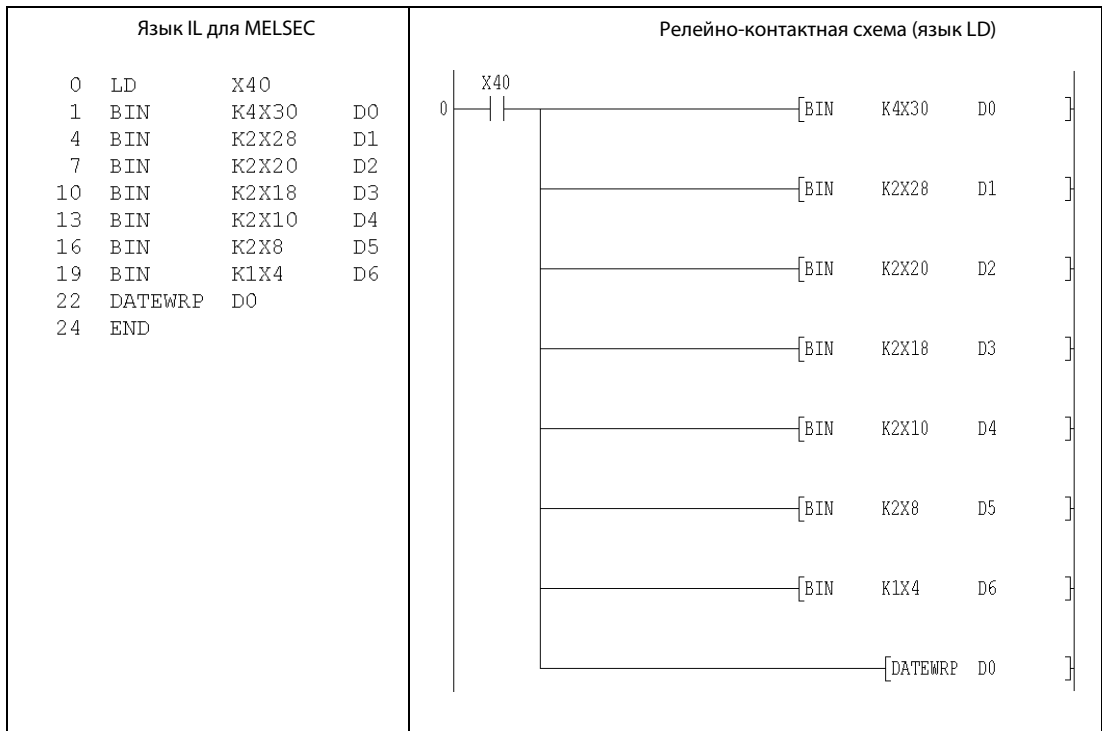
**ПРИМЕЧАНИЕ** *Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.*

**Пример 2** DATEWRP (процессор "System Q")

В следующей программе при положительном фронте сигнала X40 данные времени, поданные на входы X0...X2F в формате BCD, записываются во внутренние часы ЦПУ. Входы сопоставлены данным времени следующим образом.

- X30 – X3F = год
- X18 – X1F = час
- X28 – X2F = месяц
- X10 – X17= минута
- X20 – X27 = день
- X8 – XF = секунда

Данные времени	Год	Месяц	День	Час	Минута	Секунда	День недели
Операнд	D0	D1	D2	D3	D4	D5	D6



- 1 год
- 2 месяц, день
- 3 час, минута
- 4 секунда, день недели
- 5 данные времени

### 7.15.3 DATE+, DATE+P

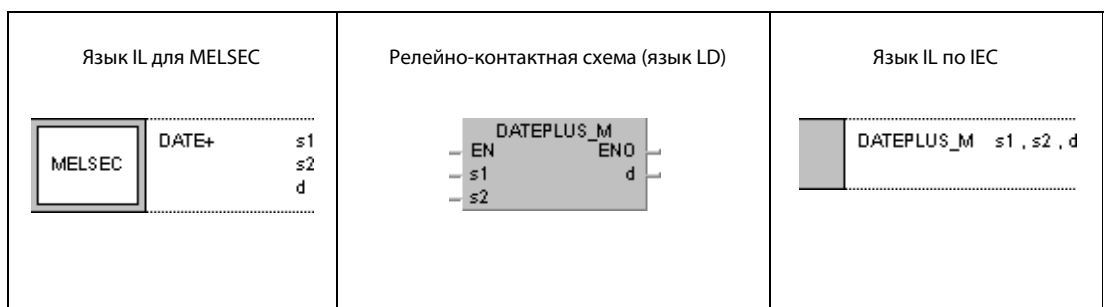
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

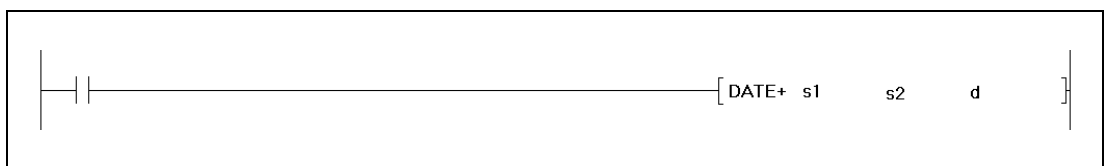
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	SMO	4	
s2	—	●	●	—	—	—	—	—			
d	—	●	●	—	—	—	—	—			

**GX IEC Developer**



**GX Developer**



**Переменные**

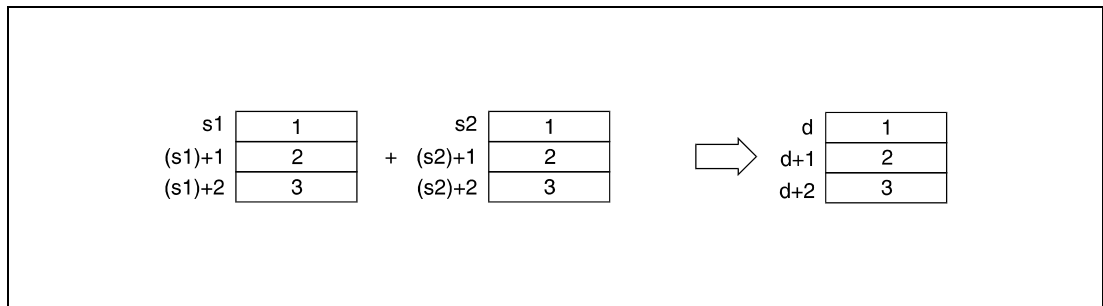
Операнд	Значение	Тип данных	
		MELSEC	IEC
s1	Слагаемые данные времени	BIN, 16 бит	массив [0..2] данных типа ANY16
s2	Слагаемые данные времени		
d	Первый адрес операнда, в котором сохраняется результат сложения данных времени.		

**Принцип действия****Сложение данных времени  
DATE+ Команда сложения**

Команда DATE+ прибавляет к данным времени в операнде, хранящемся, начиная с s1, данные времени операнда, хранящегося начиная с s2. Результат сложения сохраняется в операнде, начиная с d.

В следующей таблице указаны диапазоны данных времени в операндах с s1 + 0 по s1 + 2 (Array\_s1[1] – Array\_s1[3]), с s2 + 0 по s2 + 2 (Array\_s2[1] – Array\_s2[3]) и с d + 0 по d + 2 (Array\_d[1] – Array\_d[3]).

Данные времени	Год	Месяц	День	Час	Минута	Секунда	День недели
Диапазон	—	—	—	0 - 23	0 - 59	0 - 59	—
Операнд	—	—	—	s1 + 0 (Array_s1[0])	s1 + 1 (Array_s1[1])	s1 + 2 (Array_s1[2])	—
Операнд	—	—	—	s2 + 0 (Array_s2[0])	s2 + 1 (Array_s2[1])	s2 + 2 (Array_s2[2])	—
Операнд	—	—	—	d + 0 (Array_d[0])	d + 1 (Array_d[1])	d + 2 (Array_d[2])	—



1 час

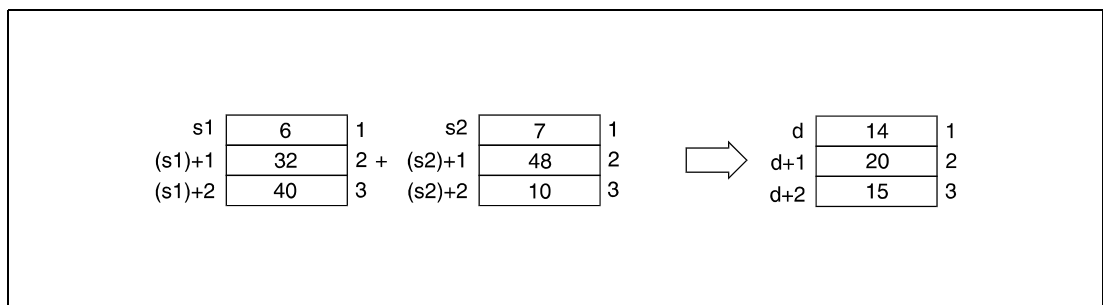
2 минута

3 секунда

На следующей иллюстрации ко времени суток

6 часов, 32 минуты и 40 секунд ((s1) + 0 – (s1) + 2, (Array\_s1[1] – Array\_s1[3])) прибавляется время суток

7 часов, 48 минут и 10 секунд ((s2) + 0 – (s2) + 2, (Array\_s2[1] – Array\_s2[3])) и результат 14 часов, 20 минут и 50 секунд сохраняется в операндах с d + 0 по d + 2 ((Array\_d[1] – Array\_d[3])).



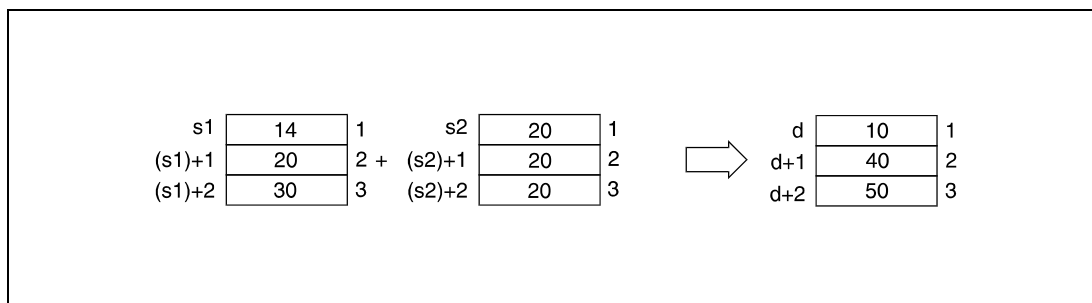
1 часы

2 минуты

3 секунды

Если результат сложения превышает 24 часа, из результата автоматически вычитаются 24 часа, чтобы получить правильное время.

В примере на следующей иллюстрации сложение времени "14 часов, 20 минут и 30 секунд" со временем "20 часов, 20 минут и 20 секунд" дает результат "34 часа, 40 минут и 50 секунд". Этот результат не может быть представлен в виде времени суток. Поэтому после вычитания 24 часов окончательный результат гласит "10 часов, 40 минут и 50 секунд" (10:40:50 на следующий день).



<sup>1</sup> час

<sup>2</sup> минута

<sup>3</sup> секунда

**ПРИМЕЧАНИЕ** *Дополнительная информация о вводе часов, минут и секунд имеется в разделе "Запись данных времени".*

**Источники ошибок** В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

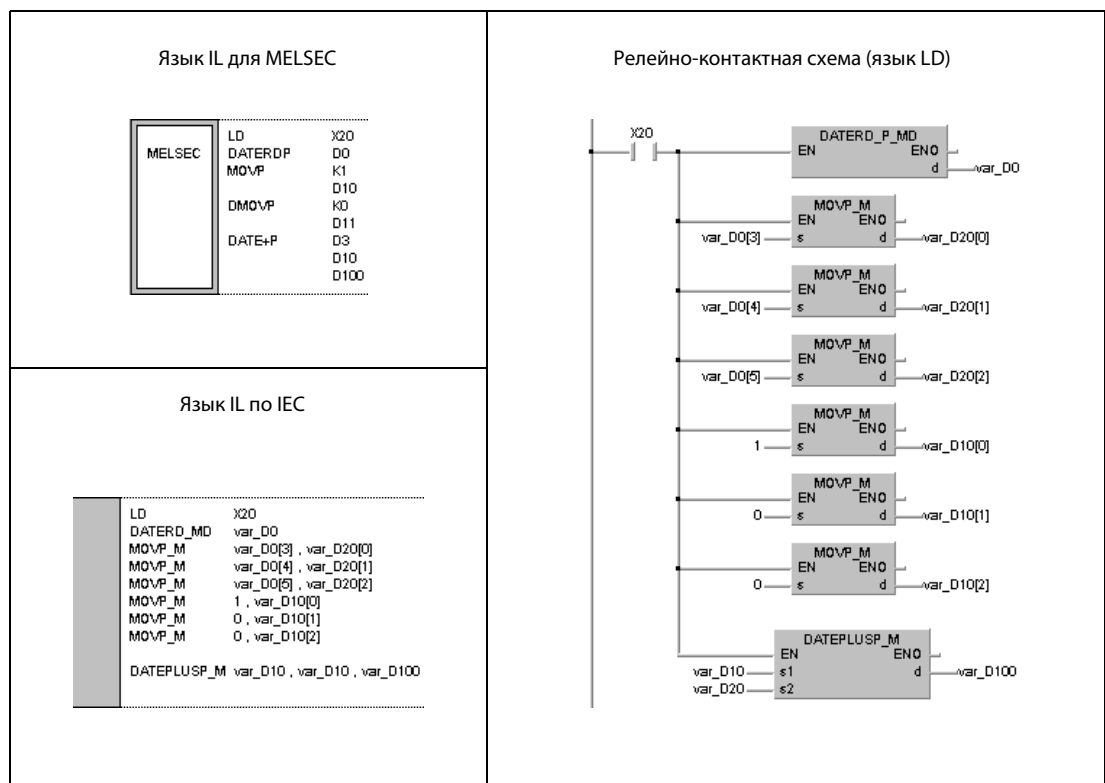
- Данные времени, указанные в операндах с (s1) + 0 по (s1) + 2 ((Array\_s1[1] – Array\_s1[3])) и с (s2) + 0 по (s2) + 2 ((Array\_s2[1] – Array\_s2[3])), находятся вне диапазона.

**Пример**

## DATE + P

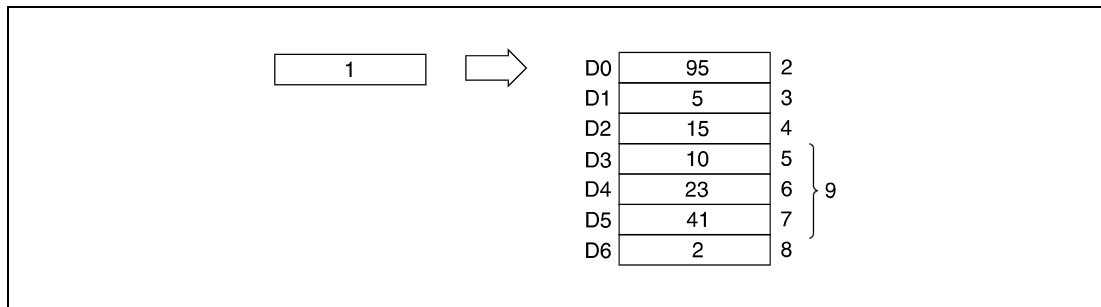
В следующей программе при положительном фронте сигнала на входе X20 данные времени считываются из внутренних часов ЦПУ с помощью команды DATERDP и сохраняются в регистрах с D0 по D6 (первая иллюстрация после примера программы). К часам, минутам и секундам этих данных с помощью команды DATE + P прибавляется один час (D10, D11, D12). Результат сохраняется в регистрах с D100 по D102 (вторая иллюстрация после примера программы).

Данные времени	Год	Месяц	День	Час	Минута	Секунда	День недели
Операнды	D0 (var_D0[0])	D1 (var_D0[1])	D2 (var_D0[2])	D3 (var_D0[3])	D4 (var_D0[4])	D5 (var_D0[5])	D6 (var_D0[6])
Операнды	—	—	—	D20 (var_D20[0])	D21 (var_D20[1])	D22 (var_D20[2])	—
Операнды	—	—	—	D10 (var_D10[0])	D11 (var_D10[1])	D12 (var_D10[2])	—
Операнды	—	—	—	D100 (var_D100[0])	D101 (var_D100[1])	D102 (var_D100[2])	—

**ПРИМЕЧАНИЕ**

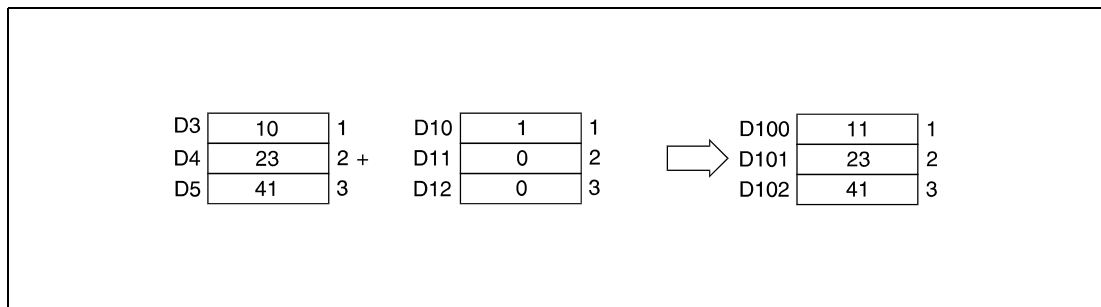
Без объявления переменных в заголовке POU этот пример программы не работоспособен. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства. Компилятор и проверяющая программа могут выдать сообщения об ошибках.

На рисунке ниже изображено считывание данных времени с помощью команды DATERDP.



- <sup>1</sup> часы процессора QnA
- <sup>2</sup> год
- <sup>3</sup> месяц (январь = 1, декабрь = 12)
- <sup>4</sup> день
- <sup>5</sup> час (24-часовой формат)
- <sup>6</sup> минута
- <sup>7</sup> секунда
- <sup>8</sup> день недели
- <sup>9</sup> данные времени суток

На рисунке ниже показано сложение, выполненное с помощью команды DATE + P.



- <sup>1</sup> час
- <sup>2</sup> минута
- <sup>3</sup> секунда

## 7.15.4 DATE-, DATE-P

## Процессор

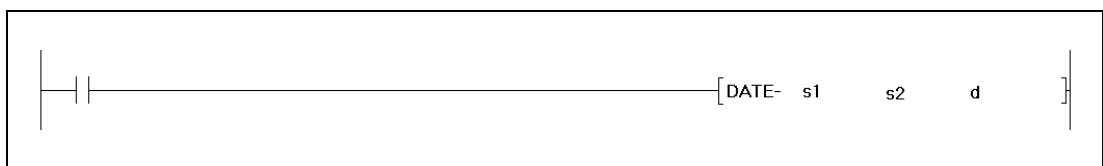
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Операнды  
MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)	Иные DY		
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	—	SMO	4
s2	—	●	●	—	—	—	—	—	—		
d	—	●	●	—	—	—	—	—	—		

GX IEC  
Developer

Язык IL для MELSEC	Релейно-контактная схема (язык LD)	Язык IL по IEC

GX  
Developer

## Переменные

Операнд	Значение	Тип данных	
		MELSEC	IEC
s1	Первый адрес операнда, в котором сохранены уменьшаемые данные времени.	BIN, 16 бит	массив [0..2] данных типа ANY16
s2	Первый адрес операнда, в котором сохранены вычитаемые данные времени.		
d	Первый адрес операнда, в котором сохраняется разность времени.		



**Принцип действия**

**Вычитание данных времени**

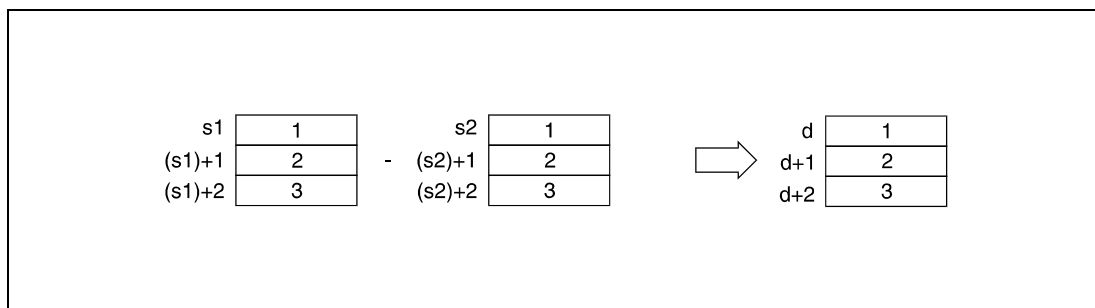
**DATE- Команда вычитания**

Команда DATE- вычитает из данных времени операнда, хранящегося начиная с s2, данные времени операнда, хранящегося начиная с s1. Разность сохраняется в операнде, начиная с d.

В следующей таблице указаны диапазоны данных, сохраняемых в операндах с (s1) + 0 по (s1) + 2 (Array\_s1[1] – Array\_s1[3]), с (s2) + 0 по (s2) + 2 (Array\_s2[1] – Array\_s2[3]) и с d + 0 по d + 2 (Array\_d[1] – Array\_d[3]).

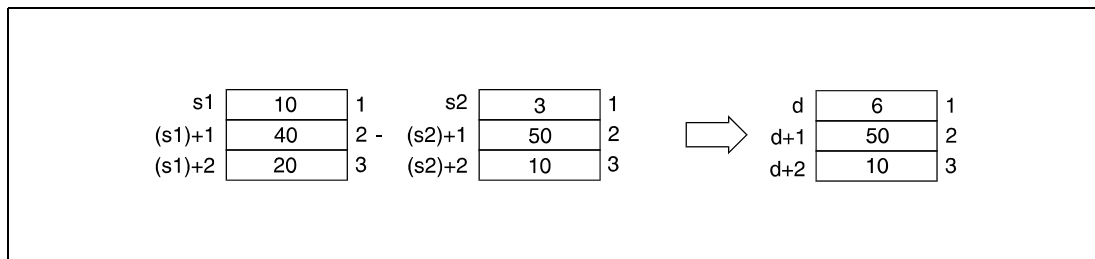
Данные времени	Год	Месяц	День	Час	Минута	Секунда	День недели
Диапазон	—	—	—	0–23	0–59	0–59	—

Данные времени	Год	Месяц	День	Час	Минута	Секунда	День недели
Операнды	—	—	—	s1 + 0 (Array_s1[0])	s1 + 1 (Array_s1[1])	s1 + 2 (Array_s1[2])	—
Операнды	—	—	—	s2 + 0 (Array_s2[0])	s2 + 1 (Array_s2[1])	s2 + 2 (Array_s2[2])	—
Операнды	—	—	—	d + 0 (Array_d[0])	d + 1 (Array_d[1])	d + 2 (Array_d[2])	—



- 1 час
- 2 минута
- 3 секунда

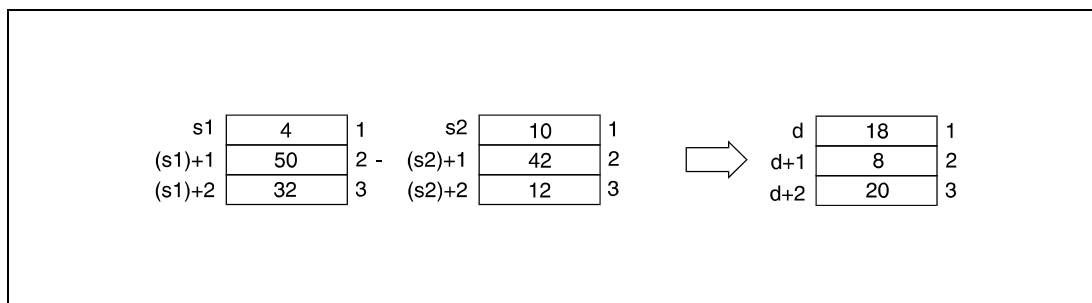
На следующей иллюстрации время "3 часа, 50 минут и 10 секунд" ((s2) + 0 – (s2) + 2 (Array\_s2[1] – Array\_s2[3])) вычитается из времени "10 часов, 40 минут и 20 секунд" ((s1) + 0 – (s1) + 2 (Array\_s1[1] – Array\_s1[3])) и результат "6 часов, 50 минут и 10 секунд" сохраняется с операндах с d + 0 по d + 2 (Array\_d[1] – Array\_d[3]).



- 1 час
- 2 минута
- 3 секунда

Если разность отрицательная, то к ней автоматически прибавляются 24 часа, чтобы получить правильное время.

В примере на следующей иллюстрации время "10 часов, 42 минуты и 12 секунд" вычитается из времени "4 часа, 50 минут и 32 секунды", что дает результат "-6 часов, 8 минут и 20 секунд". Этот результат не может быть представлен в виде времени суток. В результате прибавления 24 часов результат становится "18 часов, 8 минут и 20 секунд" (18:08:20 в предыдущий день).



<sup>1</sup> час

<sup>2</sup> минута

<sup>3</sup> секунда

**ПРИМЕЧАНИЕ** *Дополнительная информация о вводе часов, минут и секунд имеется в разделе "Запись данных времени".*

**Источники ошибок**

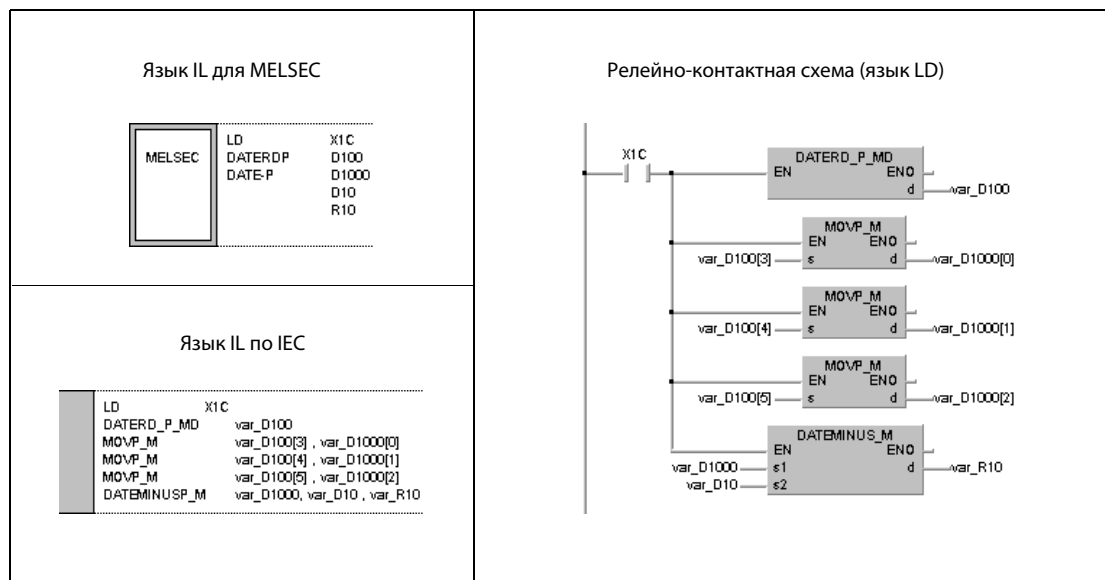
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Данные времени, указанные в операндах с (s1) + 0 по (s1) + 2 (Array\_s1[1] – Array\_s1[3]) и с (s2) + 0 по (s2) + 2 (Array\_s2[1] – Array\_s2[3]), находятся вне диапазона.

**Пример** DATE-P

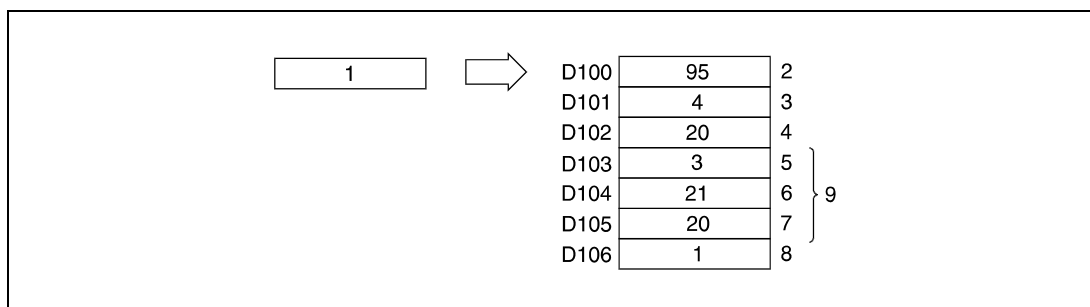
В следующей программе при положительном фронте сигнала X1C данные времени считываются из внутренних часов ЦПУ с помощью команды DATERDP и сохраняются в регистрах с D100 по D106 (первая иллюстрация после примера программы). Из времени суток, содержащихся в этих данных, с помощью команды DATE-P вычитаются 10 часов (D10), 40 минут (D11) и 10 секунд (D12). К отрицательной разности "-8 часов, 41 минута и 10 секунд" прибавляются 24 часа. Окончательный результат "16 часов, 41 минута и 10 секунд" (16:41:10 предшествующей даты) сохраняется в регистрах с R10 по R12 (вторая иллюстрация после примера программы).

Данные времени	Год	Месяц	День	Час	Минута	Секунда	День недели
Операнды	D100 (var_D100[0])	D101 (var_D100[1])	D102 (var_D100[2])	D103 (var_D100[3])	D104 (var_D100[4])	D105 (var_D100[5])	D106 (var_D100[6])
Операнды	—	—	—	D1000 (var_D1000[0])	D1001 (var_D1000[1])	D1002 (var_D1000[2])	—
Операнды	—	—	—	D10 (var_D10[0])	D11 (var_D10[1])	D12 (var_D10[2])	—
Операнды	—	—	—	R10 (var_R10[0])	R11 (var_R10[1])	R12 (var_R10[2])	—



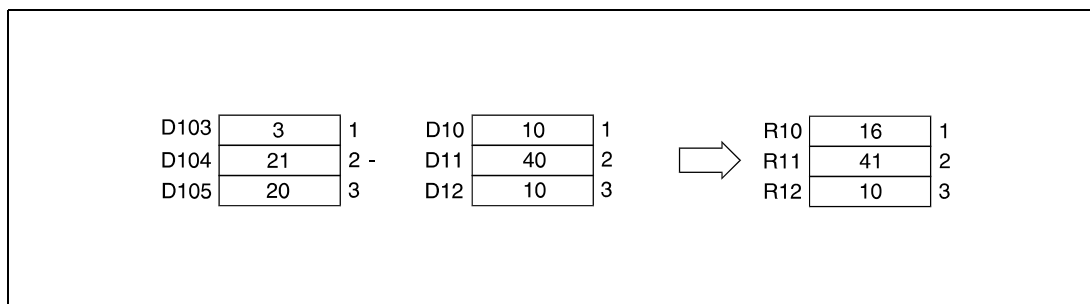
**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

На рисунке ниже изображено считывание данных времени с помощью команды DATERDP.



- 1 часы процессора QnA
- 2 год
- 3 месяц (январь = 1, декабрь = 12)
- 4 день
- 5 час (24-часовой формат)
- 6 минута
- 7 секунда
- 8 день недели
- 9 данные времени суток

На рисунке ниже изображено вычитание с помощью команды DATE-P.



- 1 час
- 2 минута
- 3 секунда

### 7.15.5 SECOND, SECONDP, HOUR, HOURP

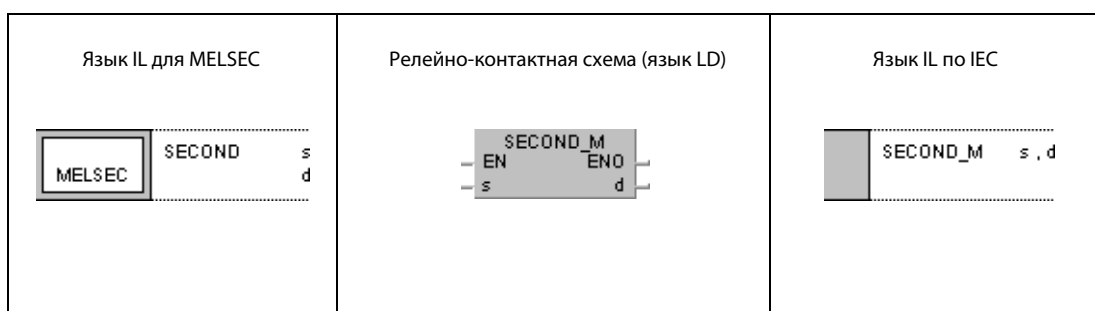
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

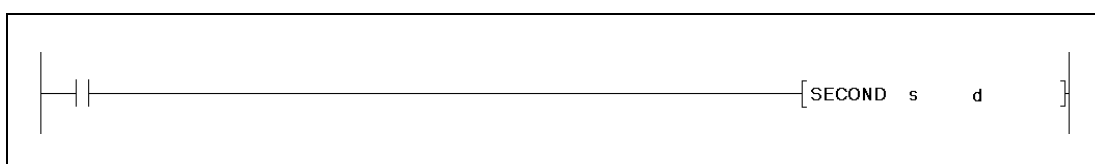
Операнды MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
SECOND											
s	—	●	●	—	—	—	—	—	—	SM0	3
d	●	●	●	●	●	●	●	—	—		
HOUR											
s	●	●	●	●	●	●	●	●	—	SM0	3
d	—	●	●	—	—	—	—	—	—		

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных	
		MELSEC	IEC
SECOND			
s	час, минута, секунда	BIN, 16/32 бита	массив [0..2] данных типа ANY16
d	секунды		ANY32
HOUR			
s	секунды	BIN, 16/32 бита	ANY32
d	час, минута, секунда		массив [0..2] данных типа ANY16

**Принцип действия**

**Преобразование формата данных времени**

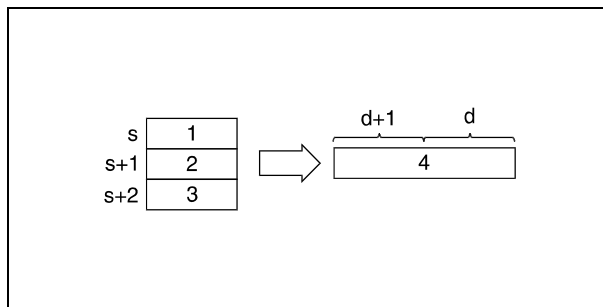
**SECOND Преобразование "часов, минут и секунд" в секунды**

Команда SECOND преобразует данные времени в формате часов, хранящиеся в операндах с s + 0 (Array\_s[1]) по s + 2 (Array\_s[3]), в формат секунд и сохраняет результат в операндах, указанных в d и d + 1.

В следующей таблице указаны диапазоны значений в операндах с s + 0 (Array\_s[1]) по s + 2 (Array\_s[3]).

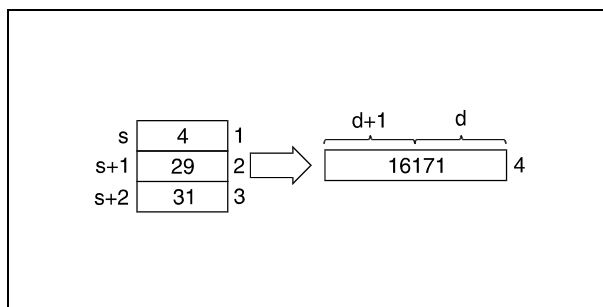
Данные времени	Год	Месяц	День	Час	Минута	Секунда	День недели
Диапазон	—	—	—	0–23	0–59	0–59	—

Данные времени	Год	Месяц	День	Час	Минута	Секунда	День недели
Операнды	—	—	—	s + 0 (Array_s[0])	s + 1 (Array_s[1])	s + 2 (Array_s[2])	—
Операнды	—	—	—	—	—	c d + 0 (Array_d[0]) по d + 1 (Array_d[1])	—



- 1 час
- 2 минута
- 3 секунда
- 4 значение, преобразованное в секунды

На следующей иллюстрации "4 часа, 29 минут и 31 секунда" преобразуются в 16171 секунду.



- 1 час
- 2 минута
- 3 секунда
- 4 значение, преобразованное в секунды

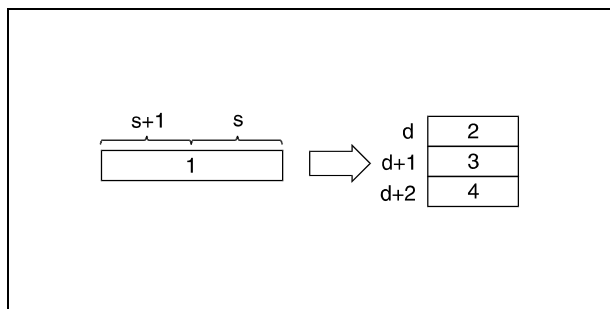
**HOUR Преобразование секунд в "часы, минуты и секунды"**

Команда HOUR преобразует данные времени в секундах, хранящиеся в операндах с s + 0 по s + 1, в формат часов.

В следующей таблице указаны диапазоны значений времени в операндах с d + 0 (Array\_d[1]) по d + 2 (Array\_d[3]).

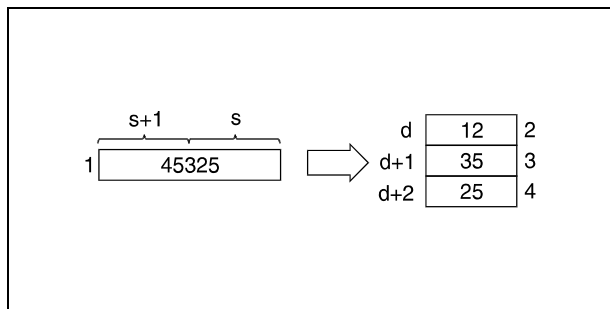
Данные времени	Год	Месяц	День	Час	Минута	Секунда	День недели
Диапазон	—	—	—	0–23	0–59	0–59	—

Данные времени	Год	Месяц	День	Час	Минута	Секунда	День недели
Операнды	—	—	—	d + 0 (Array_d[0])	d + 1 (Array_d[1])	d + 2 (Array_d[2])	—
Операнды	—	—	—	—	—	cs + 0 (Array_s[0]) по s + 1 (Array_s[1])	—



- 1 преобразуемое значение в секундах
- 2 час
- 3 минута
- 4 секунда

На следующей иллюстрации 45325 секунд преобразуются в "12 часов, 35 минут и 25 секунд".



- 1 преобразуемое значение в секундах
- 2 час
- 3 минута
- 4 секунда

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

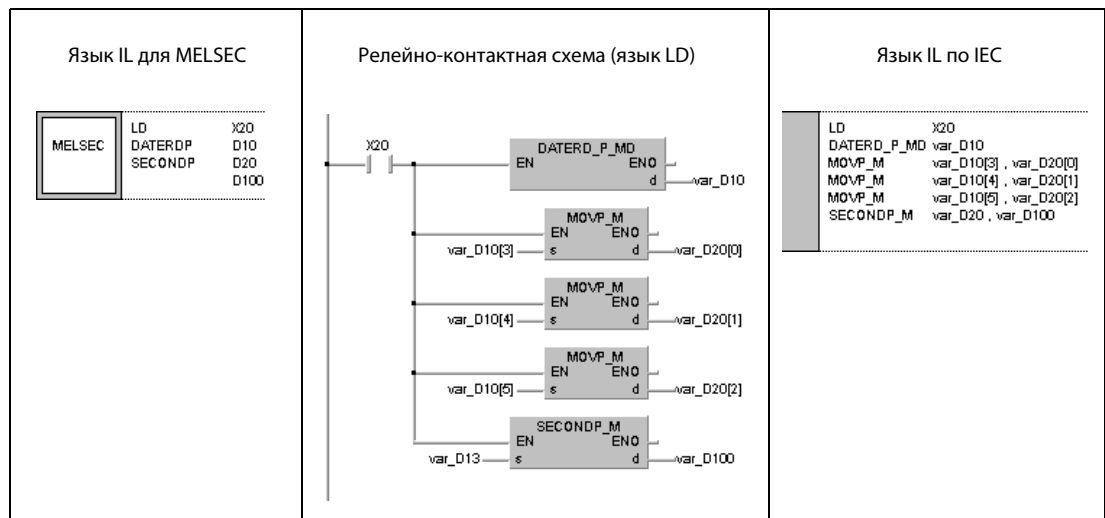
- Данные времени, указанные для команды SECOND в операндах с s + 0 (Array\_s[1]) по s + 2 (Array\_s[3]) или для команды HOUR в операндах s + 0 и s + 1, находятся вне диапазона (код ошибки 4100).

**Пример 1**

**SECONDP**

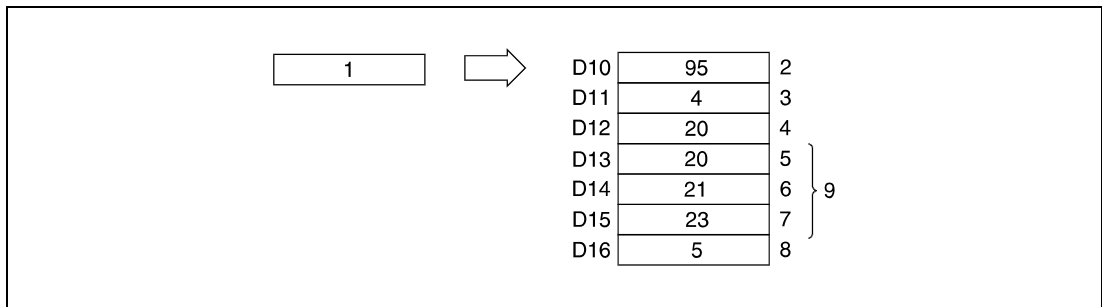
В следующей программе при положительном фронте сигнала X20 данные времени считываются из внутренних часов ЦПУ с помощью команды DATERDP и сохраняются в регистрах с D10 по D16 (первая часть иллюстрации после примера программы). Часы D20, минуты D21 и секунды D22 этих данных с помощью команды SECONDP преобразуются в секунды. Результат сохраняется в D100 и D101 (вторая часть иллюстрации после примера программы).

Данные времени	Год	Месяц	День	Час	Минута	Секунда	День недели
Операнды	D10 (var_D10[0])	D11 (var_D10[1])	D12 (var_D10[2])	D13 (var_D10[3])	D14 (var_D10[4])	D15 (var_D10[5])	D16 (var_D10[6])
Операнды	—	—	—	D20 (var_D20[0])	D21 (var_D20[1])	D22 (var_D20[2])	—
Операнды	—	—	—	—	—	от D100 (var_D10[0]) до D101 (var_D10[1])	—



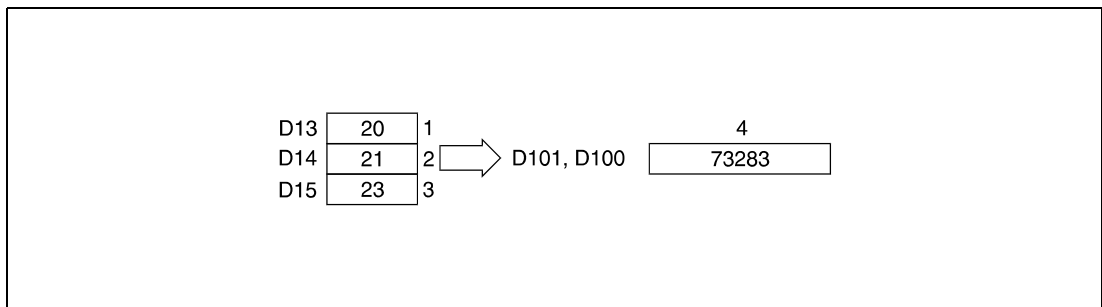


На рисунке ниже изображено считывание данных времени с помощью команды DATERDP.



- 1 часы процессора QnA
- 2 год
- 3 месяц (январь = 1, декабрь = 12)
- 4 день
- 5 час (24-часовой формат)
- 6 минута
- 7 секунда
- 8 день недели
- 9 данные времени суток

На рисунке ниже изображено преобразование в секунды с помощью команды SECONDP.

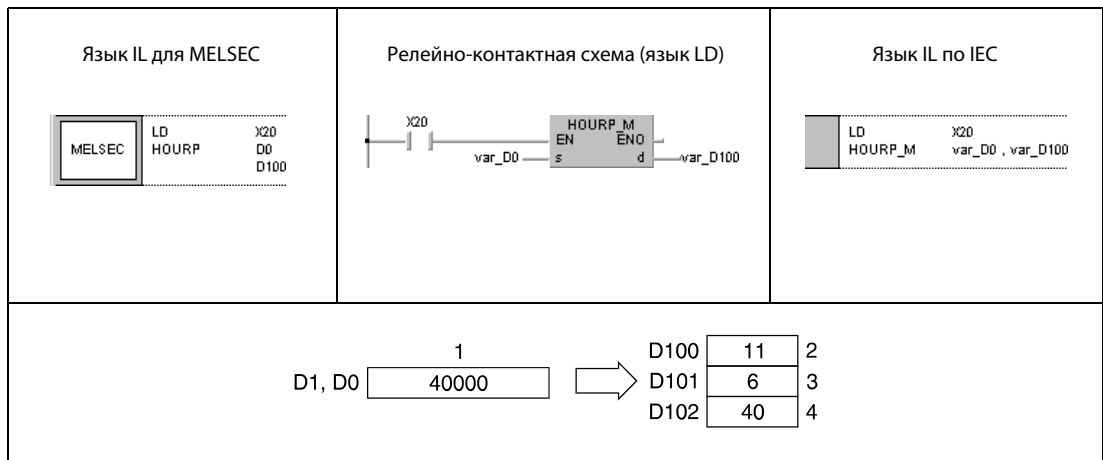


- 1 час
- 2 минута
- 3 секунда
- 4 значение, преобразованное в секунды

**Пример 2** HOURP

В следующей программе при положительном фронте сигнала X20 значение в секундах, хранящееся в D0 и D1, преобразуется в часы, минуты и секунды. Результат сохраняется в операндах, указанных в скобках.

Данные времени	Год	Месяц	День	Час	Минута	Секунда	День недели
Операнды	—	—	—	D0 (var_D0[1])	D1 (var_D0[2])	D2 (var_D0[3])	—
Операнды	—	—	—	—	—	от D100 (var_D100[0]) до D101 (var_D100[1])	—



<sup>1</sup> преобразуемое значение в секундах

<sup>2</sup> час

<sup>3</sup> минута

<sup>4</sup> секунда

**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU эти примеры программ не работоспособны. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

## 7.16 Команды для периферийных устройств

Команды для периферийных устройств позволяют выводить сообщения на периферийные устройства и вводить данные с помощью клавиатуры периферийных устройств.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту МЭК (IEC)
Вывод сообщений на периферийные устройства	MSG	MSG_M
Ввод данных с клавиатуры периферийных устройств	PKEY	PKEY_M

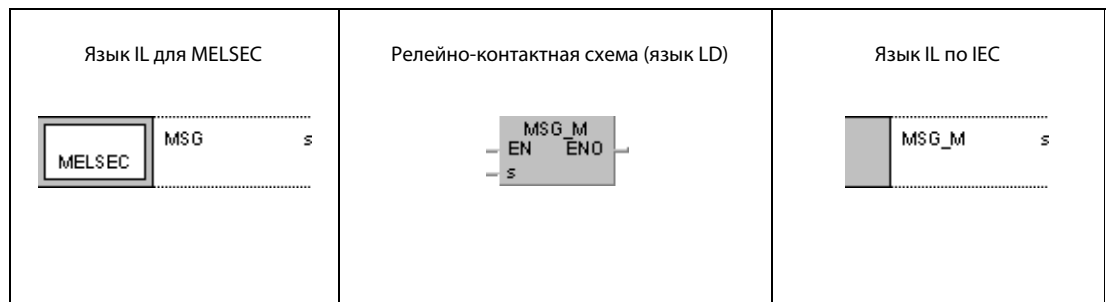
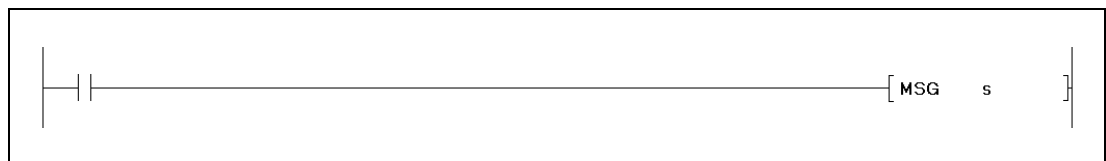
## 7.16.1 MSG

## Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Операнды  
MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□VG□	Индексные регистры Zn	Константы \$	Иные		
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	●	—	—	2

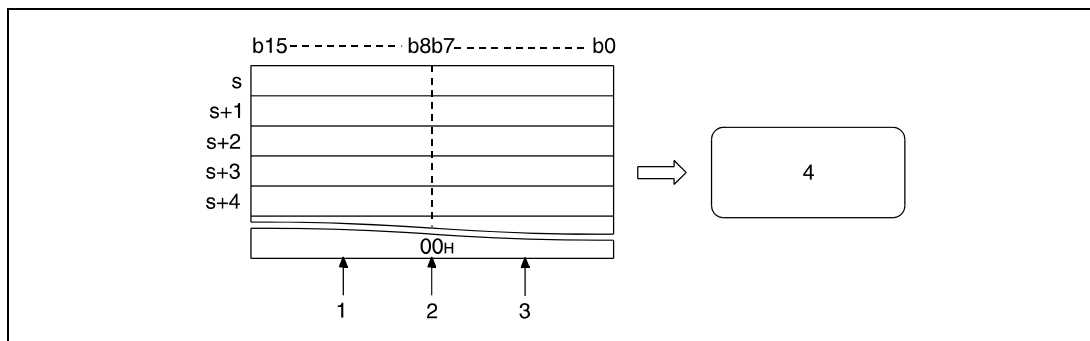
GX IEC  
DeveloperGX  
Developer

## Переменные

Операнд	Значение	Тип данных
s	Строковые данные, отображаемые на периферийном устройстве, или первый адрес операнда, в котором сохранены эти данные.	строковая величина

**Принцип действия****Вывод сообщений на периферийные устройства****MSG Команда вывода**

Команда MSG выводит строковую величину, хранящуюся в операндах, начиная с  $s$ , в качестве сообщения на периферийное устройство, указанное в режиме терминала. Конец строковой величины обозначается кодом "00н".



<sup>1</sup> 2., 4., ..., (n + 1)-ый знак

<sup>2</sup> код "00н" обозначает конец строковой величины

<sup>3</sup> 1., 3., ..., n-ый знак

<sup>4</sup> отображение строковых величин (сообщений) на периферийном устройстве

На дисплее периферийного устройства можно отобразить максимум 64 знака.

Строковые данные, указанные в  $s$ , сохраняются в регистрах диагностики с SD738 по SD773 (область памяти для сообщений).

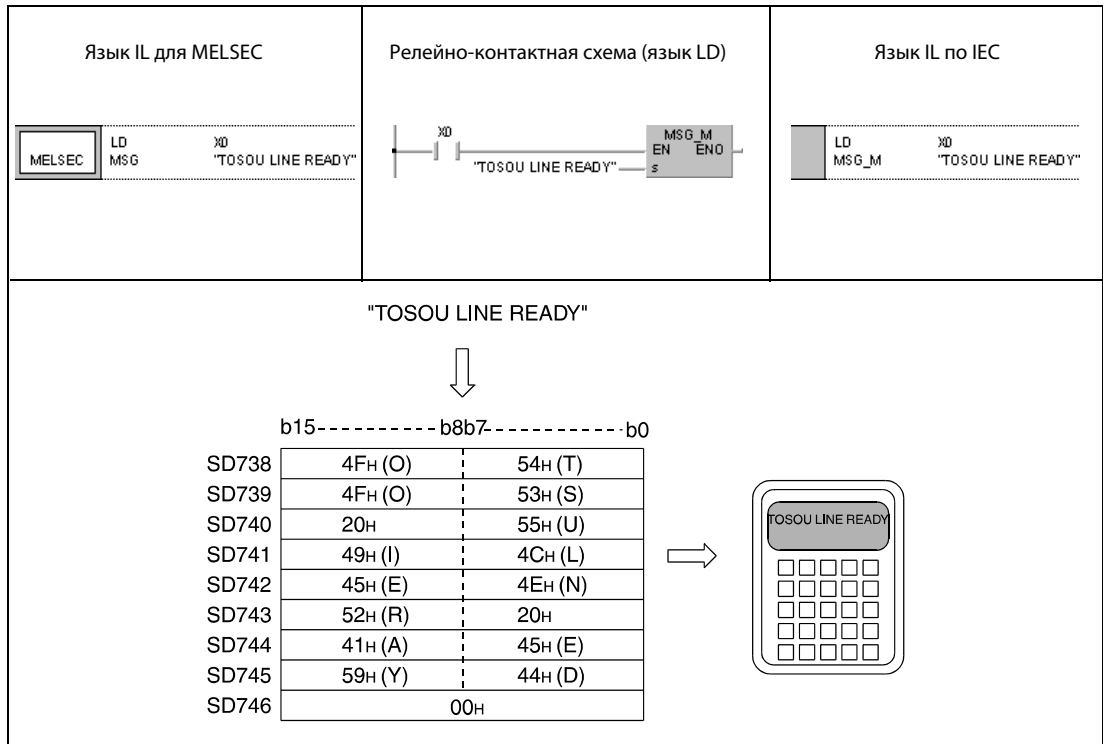
Во время выполнения команды MSG устанавливается маркер диагностики SM738 (сигнал выполнения команды MSG). Если маркер SM738 установлен, выполнение другой команды MSG не возможно.

После полного выполнения команды MSG, т. е. после отображения всех знаков на дисплее периферийного устройства, маркер диагностики SM738 сбрасывается и содержимое регистров диагностики с SD738 по SD773 (строковая величина) стирается (путем перезаписи кодом "00н").

**Пример**

MSG

В следующей программе при включении входа X0 строковая величина "TOSOU LINE READY" в качестве сообщения посылается на периферийное устройство и отображается на его дисплее.



### 7.16.2 PKEY

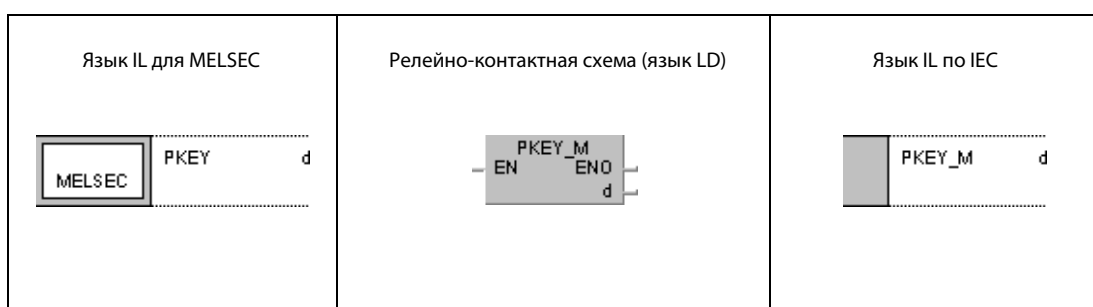
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

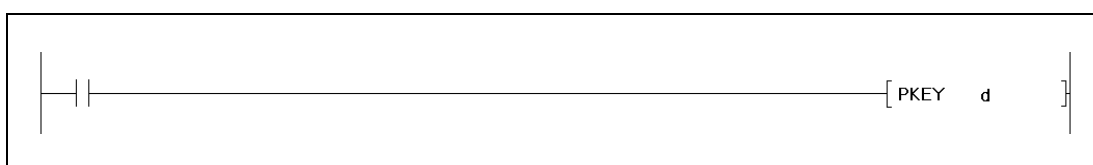
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные U
	битовые	словные		битовые	словные						
d	—	●	●	—	—	—	—	●	—	SM0	2

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
d	Первый адрес операнда, в котором сохраняется введенная строковая величина.	BIN, 16 бит

**Принцип действия****Ввод данных с клавиатуры периферийных устройств****PKEY Команда ввода**

Команда PKEY стирает слова данных в операндах, указанных с  $d + 0$  по  $d + 17$ , и устанавливает маркер диагностики SM736 (сигнал выполнения команды PKEY). Дополнительно устанавливается маркер диагностики SM737 (сигнал приема данных с клавиатуры). По окончании выполнения команды PKEY данные (знаки), которые были введены с клавиатуры периферийного устройства, определенного в режиме терминала, считываются и записываются в формате ASCII в операнды, указанные с  $d + 0$  по  $d + 17$ .

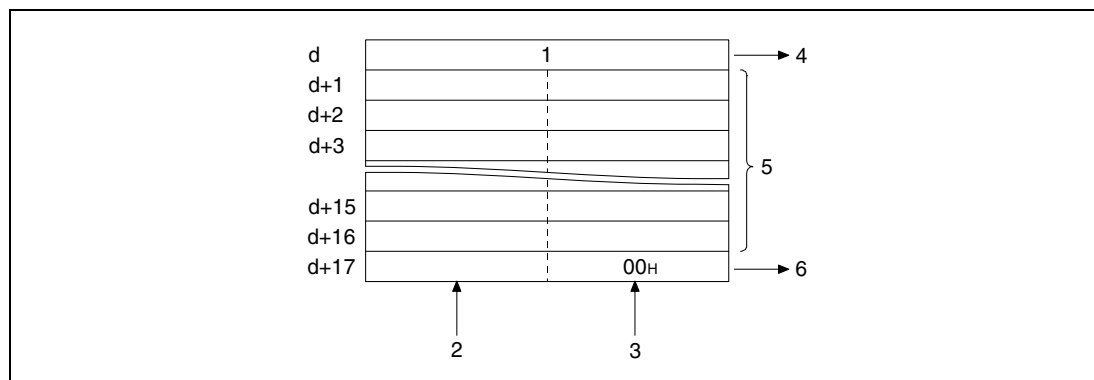
При сбросе условия выполнения команды PKEY сбрасываются и маркеры SM736 и SM737.

Маркер SM737 устанавливается при приеме знаков, введенных с клавиатуры периферийного устройства, и сбрасывается при сохранении введенных с клавиатуры данных в процессоре QnA. Если маркер SM737 установлен, прием других данных с клавиатуры периферийных устройств не возможен.

Ввод с клавиатуры периферийного устройства завершается при приеме кода "CR" (возврат каретки).

В общей сложности возможен ввод до 32 знаков. После приема 32 знаков прием данных с клавиатуры периферийного устройства прерывается, даже если код "CR" еще не был принят.

Сохранение данных (знаков), введенных с клавиатуры, в операндах, указанных с  $d + 1$  по  $d + 17$ , пояснено нижеследующей иллюстрацией.



<sup>1</sup> счетчик

<sup>2</sup> знаки со 2-го по 32-й

<sup>3</sup> знаки с 1-го по 31-й

<sup>4</sup> количество введенных знаков (двоичное значение)

<sup>5</sup> максимум 16 знаков

<sup>6</sup> Код "00H" обозначает конец введенной строковой величины (при нечетном количестве введенных знаков – в старшем байте, при четном – в младшем).

Команда PKEY не может одновременно выполняться из двух или более источников. Если команду PKEY требуется выполнять из двух или более источников, то с помощью маркера SM736 (сигнала выполнения команды PKEY) следует реализовать блокировку, предотвращающую одновременное выполнение команд.

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

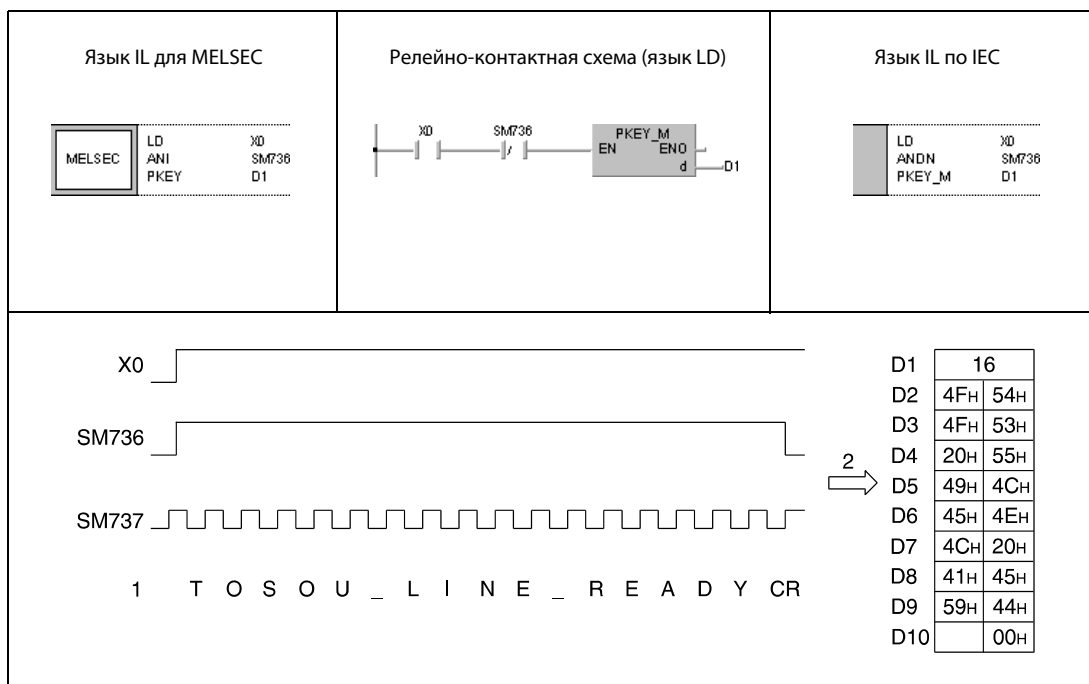
- Сделана попытка сохранить введенные с клавиатуры данные, которые находятся вне предусмотренной для сохранения области операнда, указанного с  $d + 0$  по  $d + 17$  (код ошибки 4101).



**Пример**

PKEY

В следующей программе при включении входа X0 строковая величина "TOSOU LINE READY", введенная с клавиатуры периферийного устройства, сохраняется в регистрах с D1 по D10.



<sup>1</sup> введенные данные

<sup>2</sup> сохранение введенных данных

## 7.17 Команды для управления программой

Эти команды позволяют переключать программу в различные режимы. Нижеследующая таблица содержит обзор команд.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту МЭК (IEC)
Переключение программы в резервный режим	PSTOP	PSTOP_M
	PSTOPP	PSTOPP_M
Переключение программы в резервный режим со сбросом выходов	POFF	POFF_M
	POFFP	POFFP_M
Переключение программы в режим однократного выполнения на каждый цикл	PSCAN	PSCAN_M
	PSCANP	PSCANP_M
Переключение программы в режим низкой скорости обработки	PLOW	PLOW_M
	PLOWP	PLOWP_M

**ПРИМЕЧАНИЕ** Проверьте, поддерживает ли имеющаяся у вас версия среды GX IEC Developer эти команды.

### 7.17.1 PSTOP, PSTOPP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

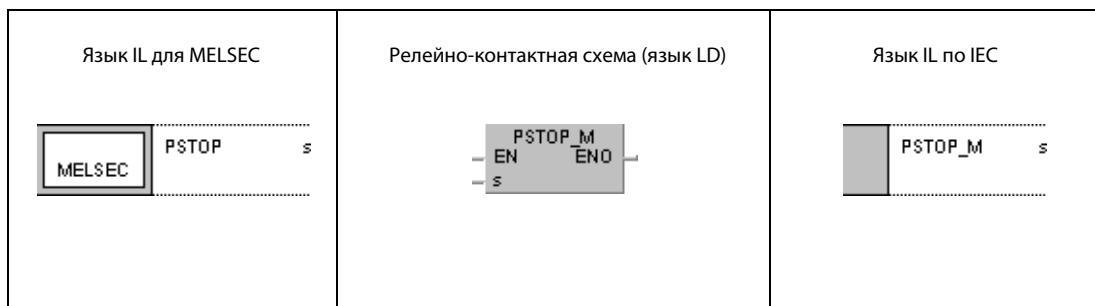
<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

**Операнды MELSEC Q**

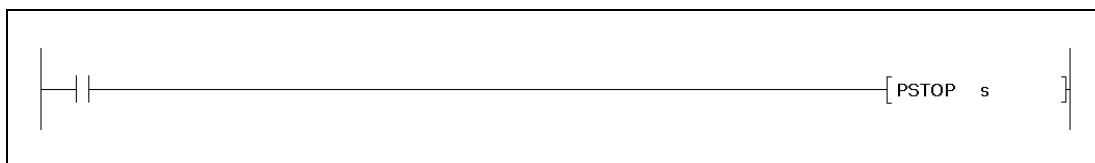
	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□\G□□	Индексные регистры Zn	Константы \$	Иные		
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	●	—	SM0	2 + n <sup>1)</sup>

<sup>1</sup> n = (количество знаков в названии программы/2) = количество дополнительных шагов (дробная часть округляется вверх)

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Строковая величина с названием программного файла, переключаемого в резервный режим, или первый адрес операнда, в котором сохранены эти данные.	строковая величина

**Принцип действия****Переключение программы в резервный режим****PSTOP Команда переключения в резервный режим**

Команда PSTOP переключает программный файл, указанный в *s*, в резервный режим. В этом режиме программный файл выполняется лишь по запросу.

Переключение в этот режим возможно только для программных файлов, сохраненных во внутренней памяти (дисконвд 0).

Переключение в резервный режим активируется лишь после обработки команды END.

Эта команда имеет приоритет обработки и в том случае, если в параметрах указан режим выполнения.

Указывать расширение .QPG в названии программного файла не требуется, так как обращение к файлу этого типа происходит автоматически.

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Программный файл с указанным названием не существует (код ошибки 2410).

**Пример****PSTOPP**

В следующей программе при положительном фронте сигнала на входе X0 программный файл "ABC" переключается в резервный режим.



### 7.17.2 POFF, POFFP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

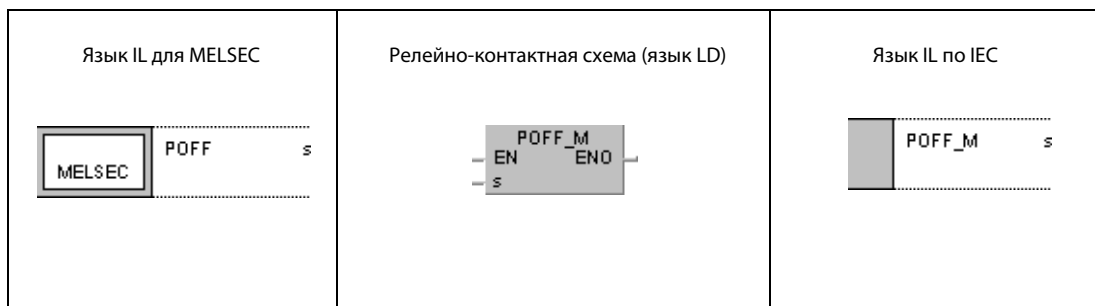
<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

**Операнды MELSEC Q**

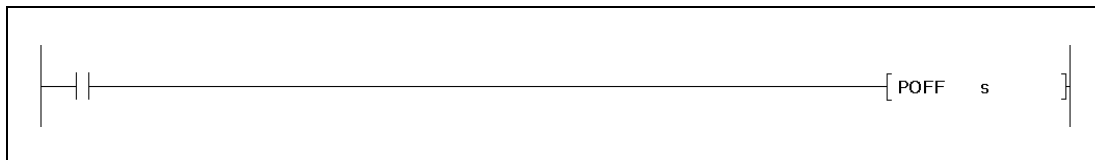
	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□□	Индексные регистры Zn	Константы \$	Иные		
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	●	—	SM0	2 + n <sup>1)</sup>

<sup>1</sup> n = (количество знаков в названии программы/2) = количество дополнительных шагов (дробная часть округляется вверх)

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором указано название программного файла, переключаемого в резервный режим со сбросом выходов.	строковая величина

**Принцип действия****Переключение программы в резервный режим со сбросом выходов****POFF, Команда переключения в резервный режим со сбросом выходов**

Команда POFF переключает программный файл, указанный в *s*, в резервный режим со сбросом выходов. В результате этого используемые программным файлом выходы и катушки устанавливаются в состояние, как если бы условия выполнения команд, управляющих этими контактами и катушками, не были выполнены. Затем происходит переключение в резервный режим.

Переключение в этот режим возможно только для программных файлов, сохраненных во внутренней памяти (на дисководе 0).

Переключение в резервный режим со сбросом выходов активируется лишь после обработки команды END.

Эта команда имеет приоритет обработки и в том случае, если в параметрах указан режим выполнения.

Расширение. QPG в названии программного файла указывать не требуется, так как обращение к файлу этого типа происходит автоматически.

**ПРИМЕЧАНИЕ**

При использовании команды POFF сбрасываются катушки, управляемые командой OUT (см. "Принцип действия").

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Программный файл с указанным названием не существует (код ошибки 2410).

**Пример****POFFP**

В следующей программе при положительном фронте сигнала X0 программный файл "ABC" переключается в резервный режим со сбросом выходов. В результате этого используемые в программном файле "ABC" выходы и катушки приводятся в состояние, как если бы условия выполнения команд, управляющих этими контактами и катушками, не были выполнены. Затем программный файл "ABC" переключается в резервный режим.



### 7.17.3 PSCAN, PSCANP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

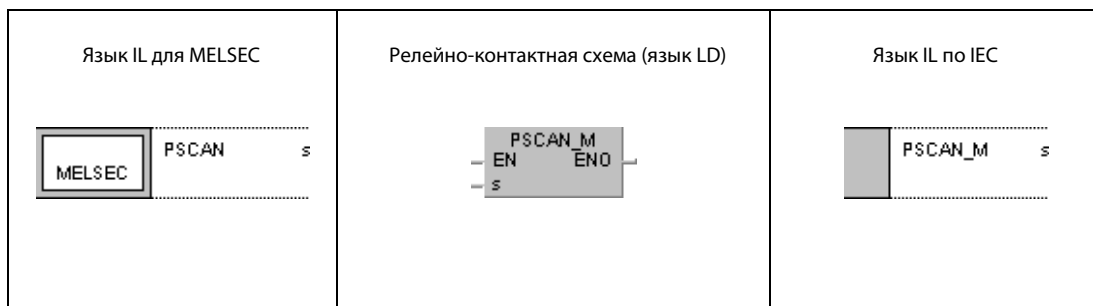
<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

**Операнды MELSEC Q**

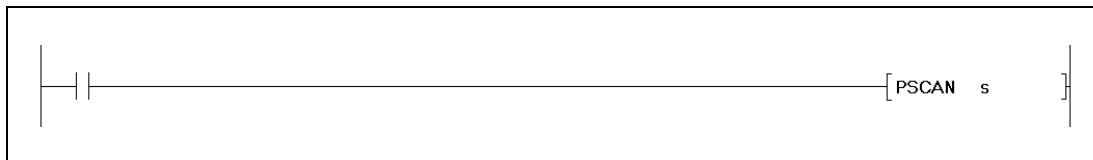
	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□\G□□	Индексные регистры Zn	Константы \$	Иные DY		
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	●	—	SM0	2 + n <sup>1)</sup>

<sup>1</sup> n = (количество знаков в названии программы/2) = количество дополнительных шагов (дробная часть округляется вверх)

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Строковая величина с названием программного файла, переключаемого в режим одного выполнения программы на цикл, или первый адрес операнда, в котором сохранены эти данные.	строковая величина

**Принцип действия****Переключение программы в режим одного выполнения программы на цикл****PSCAN Команда переключения в режим одного выполнения программы на цикл**

Команда PSCAN переключает программный файл, указанный в *s*, в режим одного выполнения программы на цикл. В этом режиме в каждом программном цикле программа выполняется один раз.

Переключение в этот режим возможно только для программных файлов, сохраненных во внутренней памяти (дисковод 0).

Переключение в режим одного выполнения программы на цикл активируется лишь после обработки команды END.

Эта команда имеет приоритет обработки и в том случае, если в параметрах указан режим выполнения.

Расширение .QPG в названии программного файла указывать не требуется, так как обращение к файлу этого типа происходит автоматически.

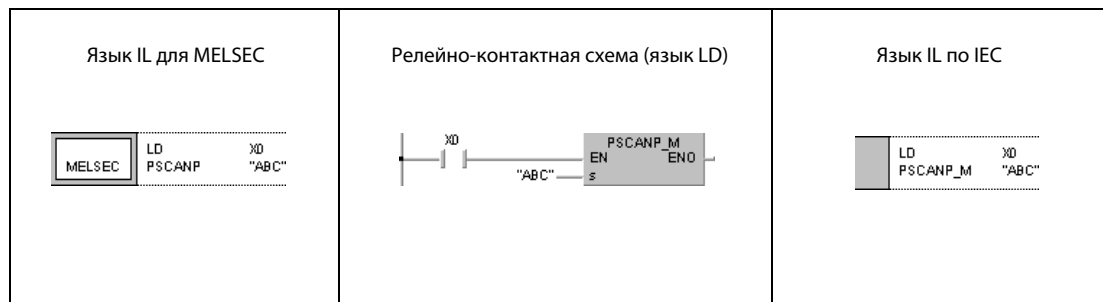
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Программный файл с указанным названием не существует (код ошибки 2410).

**Пример****PSCANP**

В следующей программе при положительном фронте сигнала на входе X0 программный файл "ABC" переключается в режим одного выполнения программы на цикл.





### 7.17.4 PLOW, PLOWP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

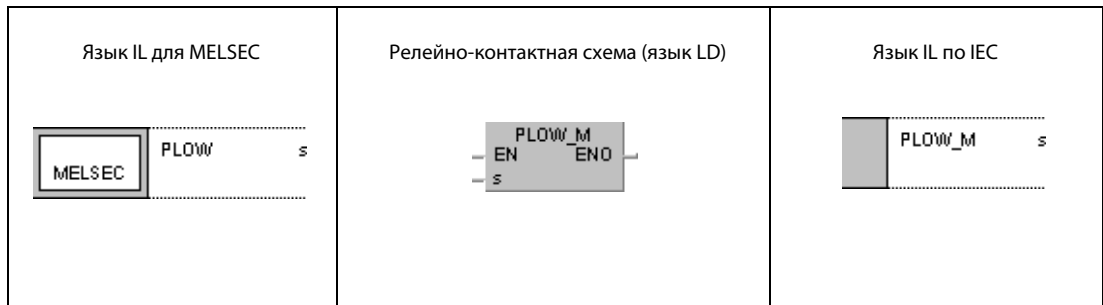
<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□□	Индексные регистры Zn	Константы \$			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	●	—	SM0	2 + n <sup>1)</sup>

<sup>1</sup> n = (количество знаков в названии программы/2) = количество дополнительных шагов (дробная часть округляется вверх)

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Строковая величина с названием программного файла, переключаемого на низкую скорость выполнения, или первый адрес операнда, в котором сохранены эти данные.	строковая величина

**Принцип действия****Переключение программы в режим низкой скорости выполнения****PLOW Команда переключения на низкую скорость выполнения**

Команда PLOW переключает программный файл, указанный в *s*, на низкую скорость выполнения. В этом режиме программа обрабатывается с низкой скоростью.

Переключение в этот режим возможно только для программных файлов, сохраненных во внутренней памяти (дисковод 0).

Переключение на низкую скорость выполнения активируется лишь после обработки команды END.

Эта команда имеет приоритет обработки и в том случае, если в параметрах указан режим выполнения.

Расширение .QPG в названии программного файла указывать не требуется, так как обращение к файлу этого типа происходит автоматически.

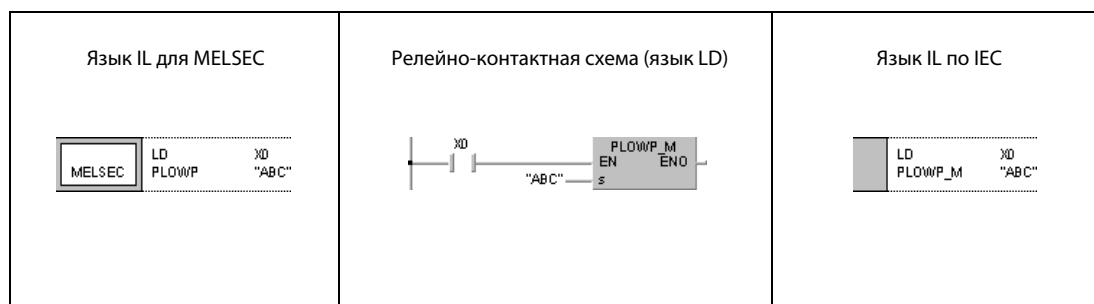
**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Программный файл с указанным названием не существует (код ошибки 2410).
- В указанном программном файле имеется команда CHK (код ошибки 4235).

**Пример****PLOWP**

В следующей программе при положительном фронте сигнала на входе X0 программный файл "ABC" переключается на низкую скорость выполнения.



## 7.18 Прочие команды

В этом разделе описаны команды для установки и сброса сторожевого таймера и флагов переноса, а также команды для задания количества циклов, записи, косвенной адресации, считывания различных данных из памяти различных видов и ввода данных в различные виды памяти. Нижеследующая таблица содержит обзор команд.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту МЭК (IEC)
Сброс сторожевого таймера	WDT	WDT_M
Установка и сброс флага переноса	STC	STC_M
	CLC	CLC_M
Задание количества циклов выполнения	DUTY	DUTY_M
Непосредственное считывание одного байта	ZRRDB	ZRRDB_M
	ZRRDBP	ZRRDBP_M
Непосредственная запись одного байта	ZRWRB	ZRWRB_M
	ZRWRBP	ZRWRBP_M
Сохранение операнда для косвенного обращения	ADRSET	ADRSET_M
	ADRESETP	ADRESETP_M
Ввод числовых значений с клавиатуры	KEY	KEY_MD
Сохранение содержимого индексных регистров в регистре	ZPUSH	ZPUSH_M
	ZPUSHP	ZPUSHP_M
Восстановление содержимого индексных регистров из регистра	ZPOP	ZPOP_M
	ZPOPP	ZPOPP_M
Запись данных в регистр EEPROM	EROMWR	EROMWR_M
	EROMWRP	EROMWRP_M

**ПРИМЕЧАНИЕ** Команды *ADRSET* и *ADRESETP* можно использовать только при программировании в среде *GX Developer*.

### 7.18.1 WDT, WDTP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	● <sup>1</sup>	●	●

<sup>1</sup> AnA + MELSECNET/10

Операнды MELSEC A

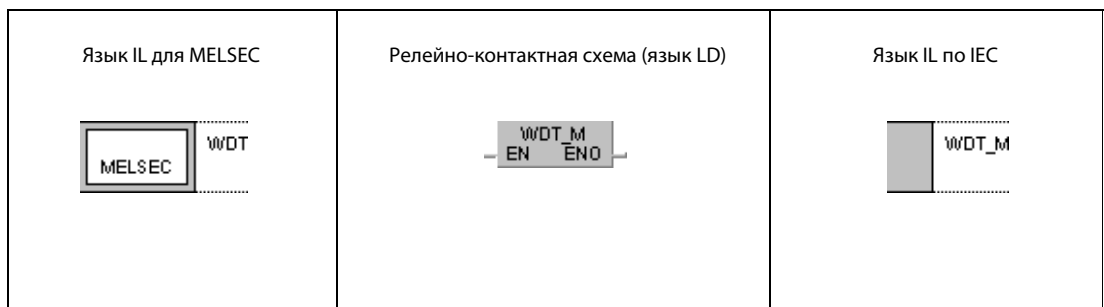
Операнды																	Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки		
битовые							словные (16 бит)							Константы		Указатели				Уровень	M9012	M9010 M9011	
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K				H (16#)	P	I	N
																					● <sup>1</sup>		

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU (AnA + MELSECNET/10) можно найти в 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

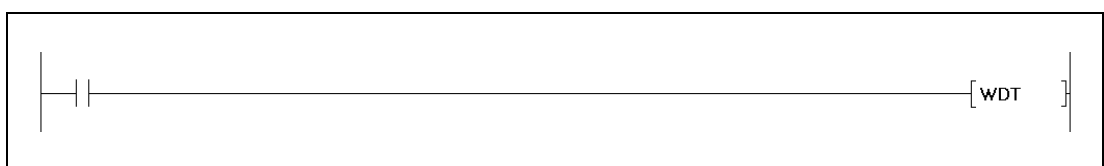
Операнды MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы	Иные			
битовые	словные		битовые	словные							
—	—	—	—	—	—	—	—	—	—	—	1

GX IEC Developer



GX Developer



Переменные

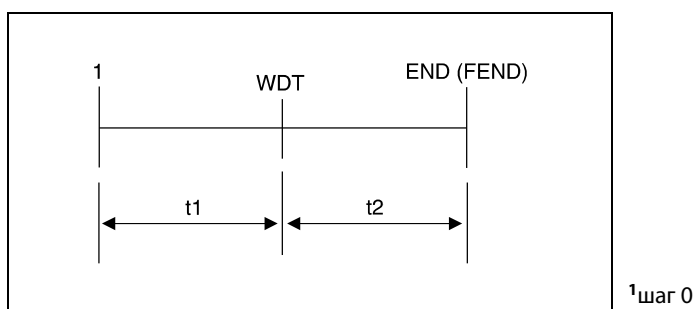
Операнд	Значение	Тип данных
—	—	—

**Принцип действия**      **Сброс сторожевых таймеров**  
**WDT      Сброс**

Команда WDT сбрасывает сторожевой таймер (WDT) в основной программе.

Программирование этой команды имеет смысл только в том случае, если при определенных условиях длительность цикла основной программы (между шагом 0 и командой END/FEND) превышает заданный интервал сторожевого таймера. Если длительность цикла превышает заданное значение сторожевого таймера в любом программном цикле, то вместо использования этой команды следует соответственно изменить настройку параметра сторожевого таймера.

Настройку сторожевого таймера необходимо выбрать так, чтобы ни интервал  $t_1$  (между шагом 0 и командой WDT), ни интервал  $t_2$  (между командами WDT и END/FEND) не превышали заданное значение.



Команду WDT можно применять в программном цикле сколько угодно часто. Однако при написании программы следует учитывать, что в случае ошибки выходы не отключаются непосредственно.

Команда WDT не стирает значения времени программного цикла, сохраненные в регистрах. Поэтому сохраненные в регистрах значения при некоторых обстоятельствах могут превышать настройку таймера WDT, установленную с помощью параметра.

**ПРИМЕЧАНИЕ**      *В центральных процессорах серий A3H, A3M, AnA, AnAS и AnU (серия "A") сторожевой таймер имеет фиксированную настройку, которая не может быть изменена.*

### 7.18.2 STC, CLC

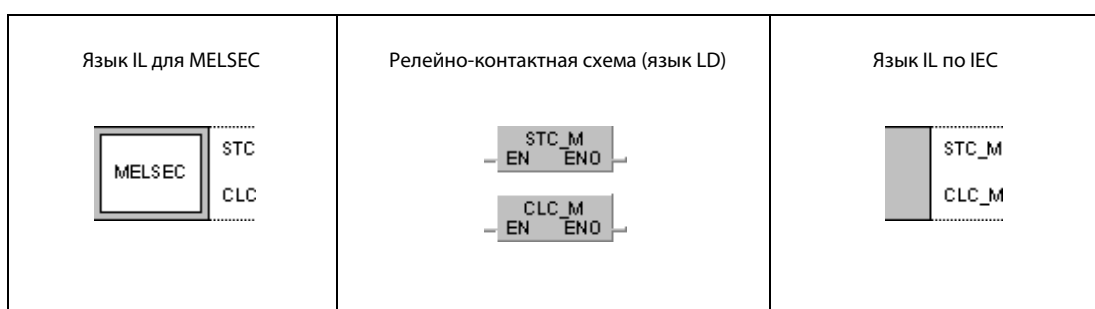
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

**Операнды MELSEC A**

Операнды																			Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9010 M9011			
битовые						словные (16 бит)						Константы		Указатели		Уровень										
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V	K	H (16#)	P						I	N	
																							1			

**GX IEC Developer**



**Переменные**

Операнд	Значение	Тип данных
—	—	—

**Принцип действия**

**Установка и сброс флага переноса**

**STC Установка флага переноса**

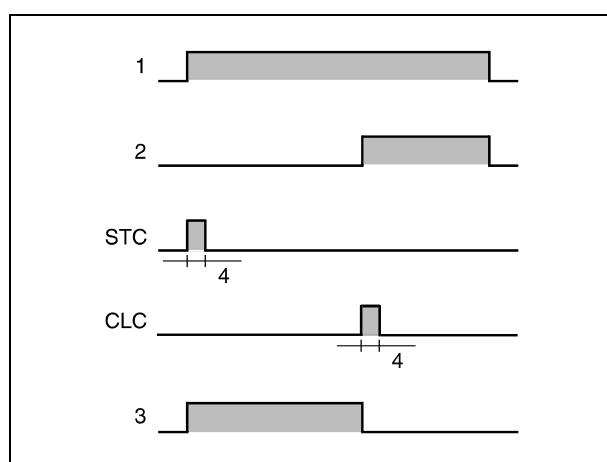
При процессах вращения и сдвига битов во флаге переноса сохраняется переносимая цифра (0 или 1). В программе перенос отображается в виде контакта – с помощью специального маркера M9012. Маркер M9012 установлен, если флаг переноса равен 1, и не установлен, если флаг переноса равен 0.

При выполнении команды STC флаг переноса (M9012) устанавливается принудительно.

**CLC Сброс флага переноса**

В результате выполнения команды CLC сбрасывается флаг переноса. Одновременно сбрасывается специальный маркер M9012.

Команды STC/CLC выполняются один раз при нарастающем фронте сигнала входного условия.

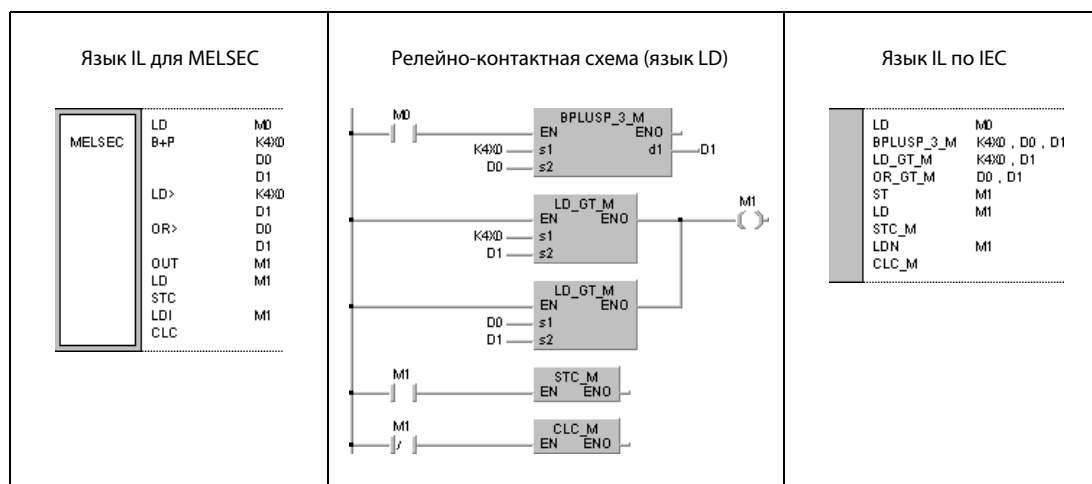


- 1 условие выполнения команды STC
- 2 условие выполнения команды CLC
- 3 флаг переноса (M9012)
- 4 выполняется один раз

**Пример**

**STC, CLC**

В следующей программе при положительном фронте сигнала M0 двоично-десятичные данные входов X0...XF прибавляются к двоично-десятичным данным D0 и результат сохраняется в D1. Если сумма больше 9999, устанавливается маркер M1 и выполняется команда STC (устанавливается маркер M9012). Если результат меньше или равен 9999, флаг переноса не устанавливается.



**7.18.3 DUTY**

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	●

**Операнды MELSEC A**

	Операнды															Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9011						
	битовые					словные (16 бит)					Константы		Указатели	Уровень												
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V	K	H (16#)	P	I	N
n1																	●	●								
n2																	●	●								●
d		●																					● <sup>1</sup>	● <sup>2</sup>		

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

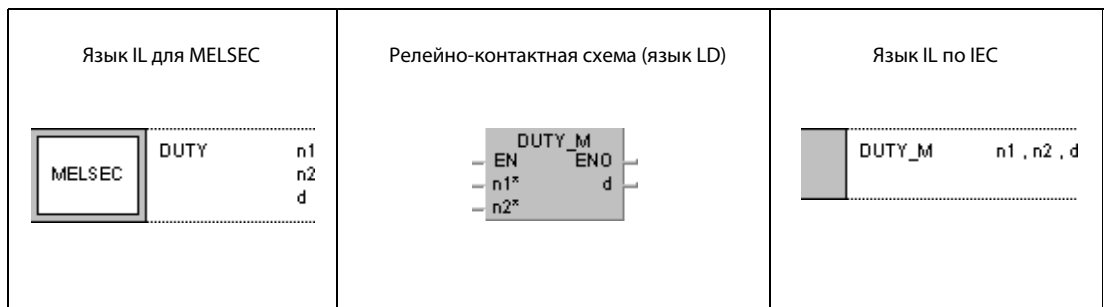
<sup>2</sup> Индексация возможна только в процессорах A3H, A3M, AnA, AnAS и AnU.

**Операнды MELSEC Q**

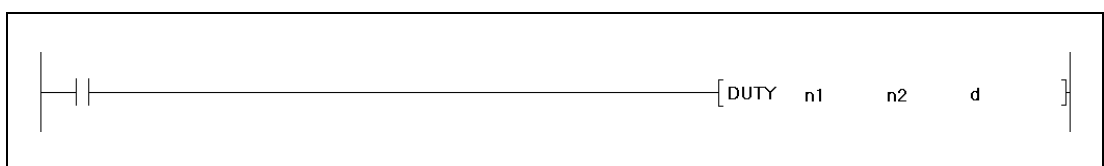
	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
n1	●	●	●	●	●	●	●	●	SM0	4	
n2	●	●	●	●	●	●	●	●			
d	● <sup>1</sup>	—	—	—	—	—	—	—			

<sup>1</sup> только SM420...SM424 и SM430...SM434

**GX IEC Developer**



**GX Developer**



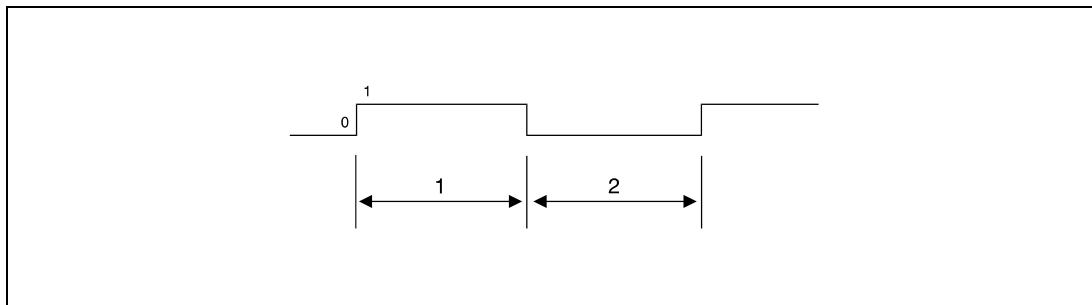
**Переменные**

Операнд	Значение	Тип данных
n1	Количество циклов, в которых установлены специальные маркеры или маркеры диагностики.	BIN, 16 бит
n2		
d	Адреса специальных маркеров и маркеров диагностики (серия "A" = M9020 – M9024, серия "Q"/"System" Q = SM420 – SM424 и SM430 – SM434).	бит



**Принцип действия****Задание количества циклов выполнения операнда****DUTY Задание количества циклов выполнения**

Команда DUTY включает операнды, указанные в d, (серия "A" = M9020...M9024, серия "Q"/"System Q" = SM420...SM424 и SM430...SM434), на количество программных циклов, указанное в n1, и отключает их на количество программных циклов, указанное в n2. После этого один или несколько соответствующих маркеров (серия "A" = специальные маркеры, серия "Q"/"System Q" = маркеры диагностики) можно использовать в качестве входного условия для последующих операций.



<sup>1</sup> количество программных циклов с выполнением

<sup>2</sup> количество программных циклов без выполнения

Программы, выполняемый один раз за цикл, используют маркеры с SM420 по SM424 (серия "Q" и "System Q").

Программы с низкой скоростью обработки используют маркеры с SM430 по SM434 (серия "Q" и "System Q").

В начале обработки (состояние инициализации) маркеры выключены (серия "A" = с M9020 по M9024, серия "Q"/"System Q" = с SM420 по SM424 и с SM430 по SM434).

Если  $n1 = 0$ , маркеры остаются выключенными.

Если  $n2 = 0$  и значение в n1 больше 0, маркеры устанавливаются и остаются включенными.

Значения в n1, n2 и d устанавливаются при вызове команды DUTY. Импульс цикла (маркер) включается или выключается при достижении команды END.

**Источники ошибок**

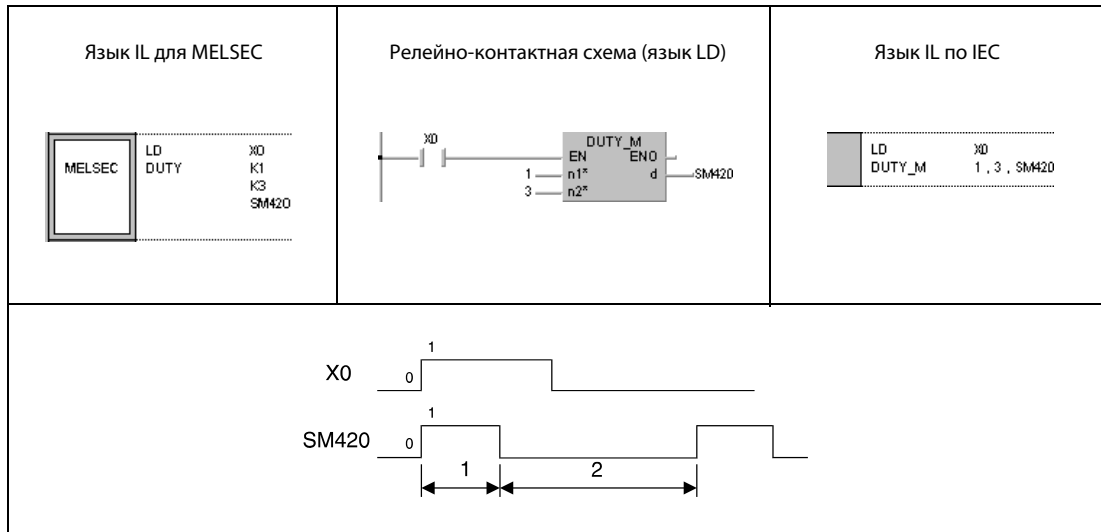
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В случае серии "A" указанный в d операнд не является маркером из диапазона M9020...M9024, а в случае серии "Q"/"System Q" – маркером из диапазона SM420...SM424 или SM430...SM434 (код ошибки 4101).
- В n1 и n2 указаны отрицательные значения (код ошибки 4100).

**Пример**

DUTY (серия "Q")

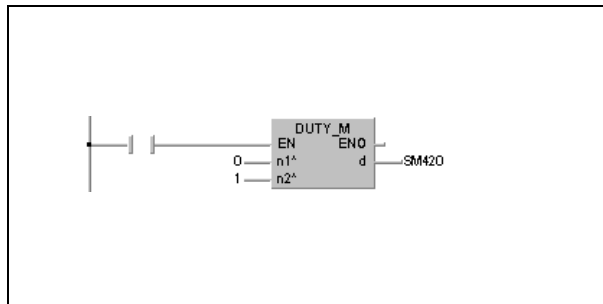
В следующей программе при положительном фронте сигнала X0 маркер диагностики устанавливается на один программный цикл и сбрасывается на 3 цикла. Этот процесс повторяется на протяжении всей работы программы (см. примечание ниже).



- <sup>1</sup> один программный цикл с выполнением
- <sup>2</sup> три программных циклах без выполнения

**ПРИМЕЧАНИЕ**

После сброса условия выполнения (X0 выключен) импульс включения, вырабатываемый командой DUTY, (и, тем самым, циклическая установка и сброс указанного маркера) продолжает действовать. Чтобы прервать дальнейший вывод импульса включения, необходимо вставить в программу следующий фрагмент.



### 7.18.4 ZRRDB, ZRRDBP

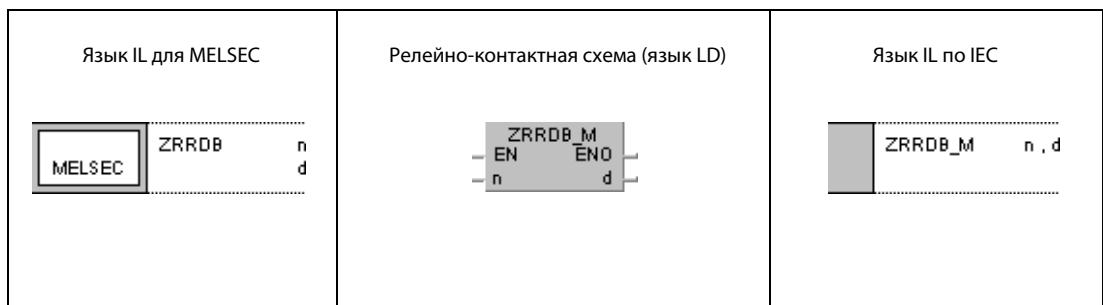
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

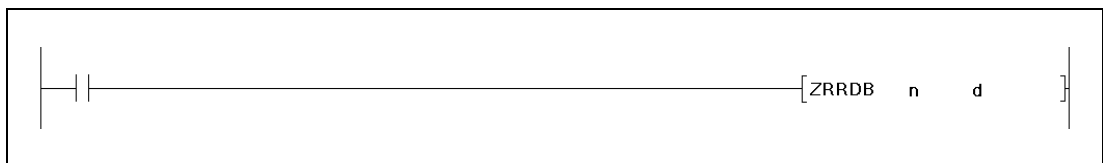
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
n	●	●	●	●	●	●	●	●	—	SM0	3
d	●	●	●	●	●	●	●	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

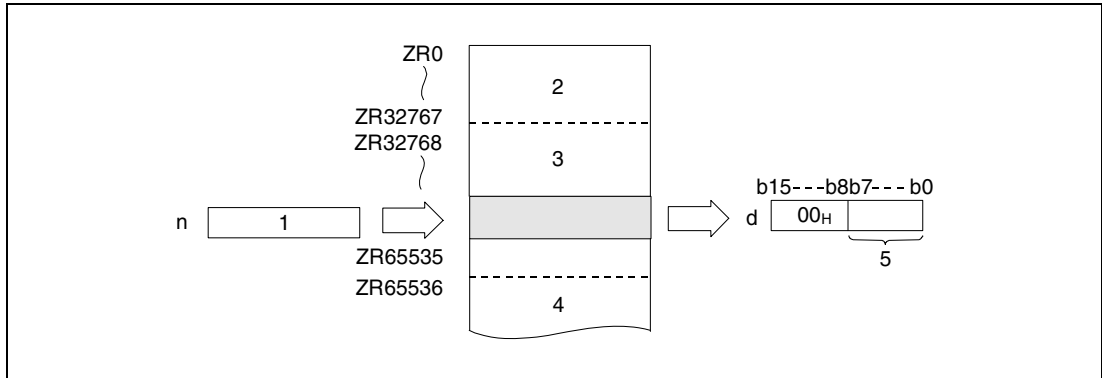
Операнд	Значение	Тип данных
n	Последовательный адрес байта, считываемого из регистра файлов.	BIN, 32 бита
d	Операнд, в котором сохраняются считанные байты.	BIN, 16 бит

**Принцип действия**

**Непосредственное считывание одного байта из регистра файлов**

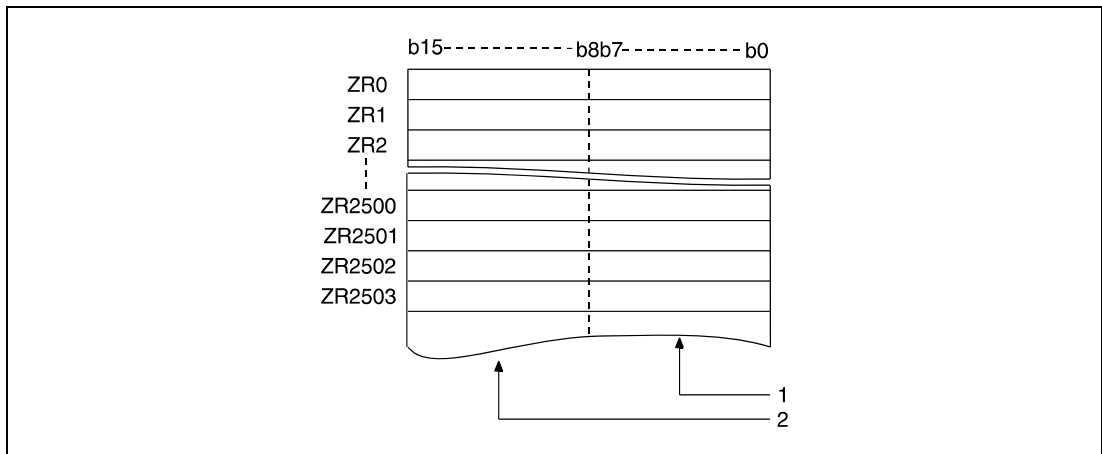
**ZRRDB Считывание одного байта**

Команда ZRRDB считывает байт из указанного в n адреса регистра файлов. При адресации байта не указывается адрес блока. Считанный байт сохраняется в младшем байте операнда, указанного в d. В старший байт операнда, указанного в d, записывается значение "00н".



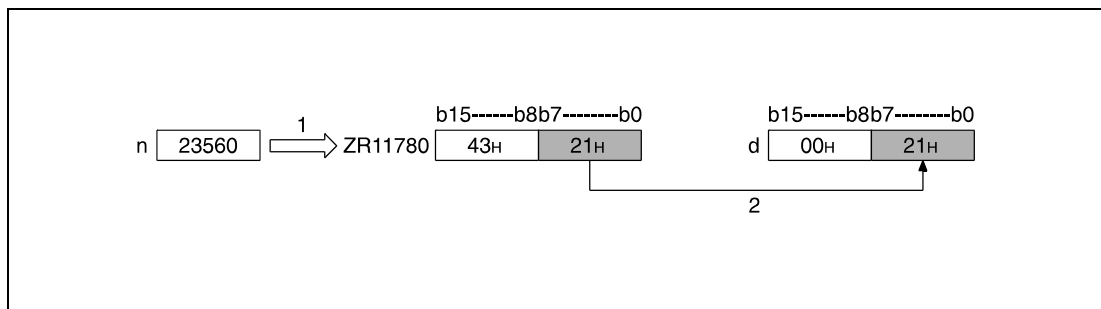
- 1 последовательный адрес байта
- 2 область регистра файлов для блока 0
- 3 область регистра файлов для блока 1
- 4 область регистра файлов для блока 2
- 5 считанный байт

Взаимосвязь между адресами регистров файлов и соответствующими последовательными адресами байтов показана на следующей иллюстрации.



- 1 область памяти для четных адресов байтов (здесь: адреса от 0 до 5006)
- 2 область памяти для нечетных адресов байтов (здесь: адреса от 1 до 5007)

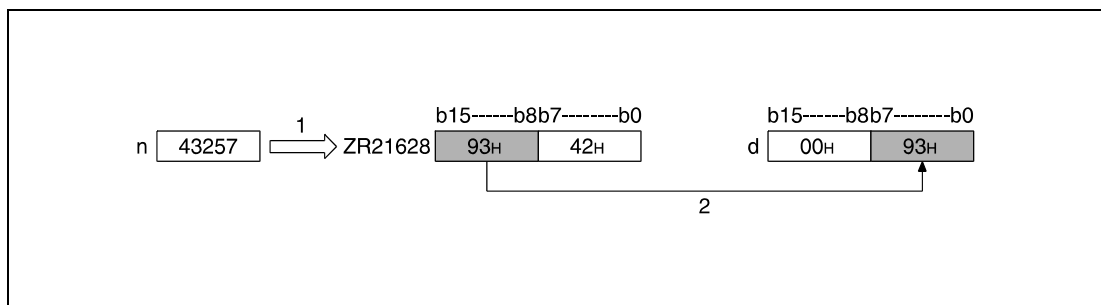
Например, при указании адреса 23560 считывается младший байт регистра файлов ZR11780.



<sup>1</sup> адресация

<sup>2</sup> сохранение

Например, при указании адреса 43257 считывается младший байт регистра файлов ZR21628.



<sup>1</sup> адресация

<sup>2</sup> сохранение

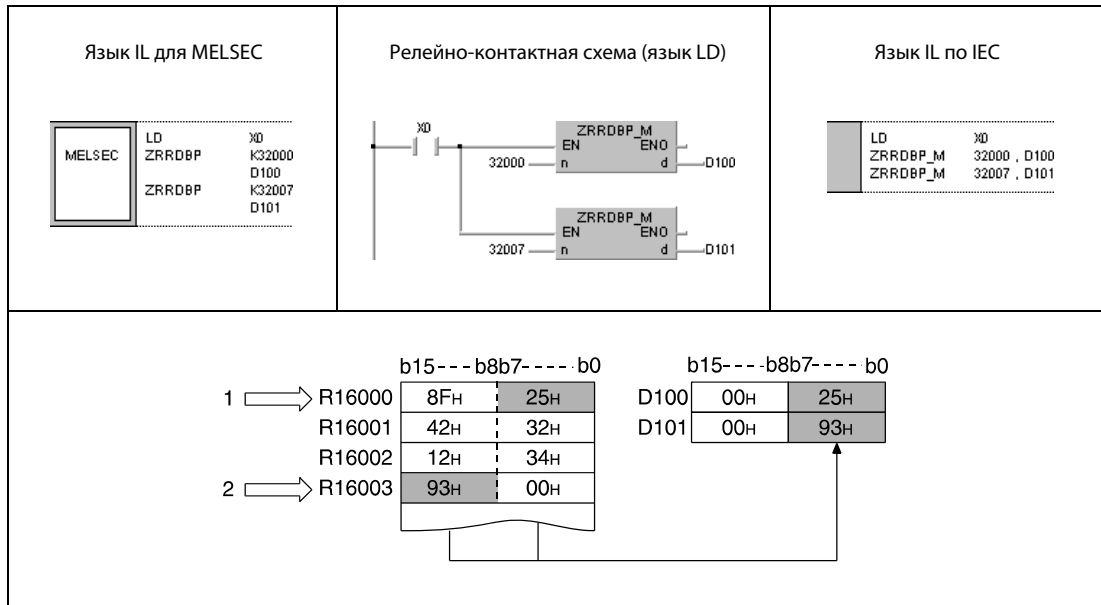
### Источники ошибок

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанный в n адрес операнда (последовательный адрес байта) находится вне области, предусмотренной для сохранения (код ошибки 4101).

**Пример** ZRRDBP

В следующей программе при положительном фронте сигнала X0 считываются младший байт регистра файлов R16000 (адрес байта 32000) и старший байт регистра файлов R16003 (адрес байта 32007). Содержимое этих байтов сохраняется в регистрах D100 и D101.



<sup>1</sup> последовательный адрес байта 32000 (младший байт регистра файлов R16000)

<sup>2</sup> последовательный адрес байта 32007 (старший байт регистра файлов R16003)

### 7.18.5 ZRWRB, ZRWRBP

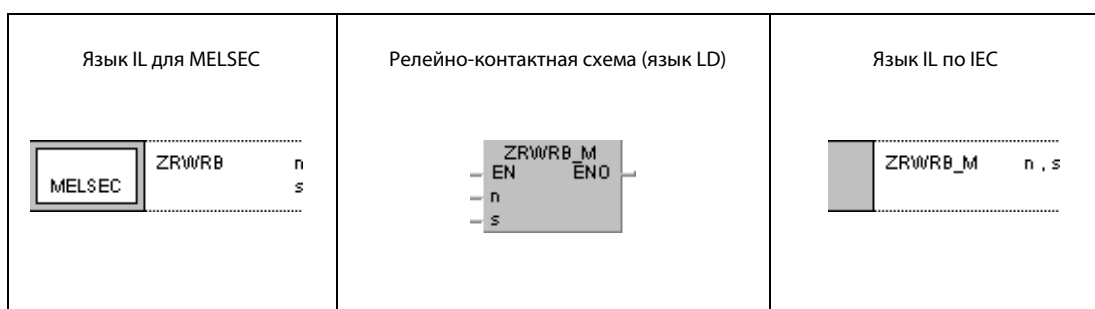
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

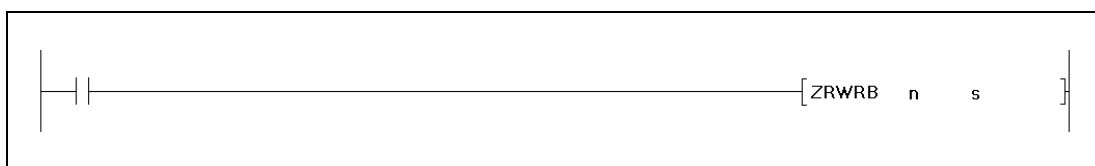
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
n	●	●	●	●	●	●	●	●	—	SM0	3
s	●	●	●	●	●	●	●	●	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

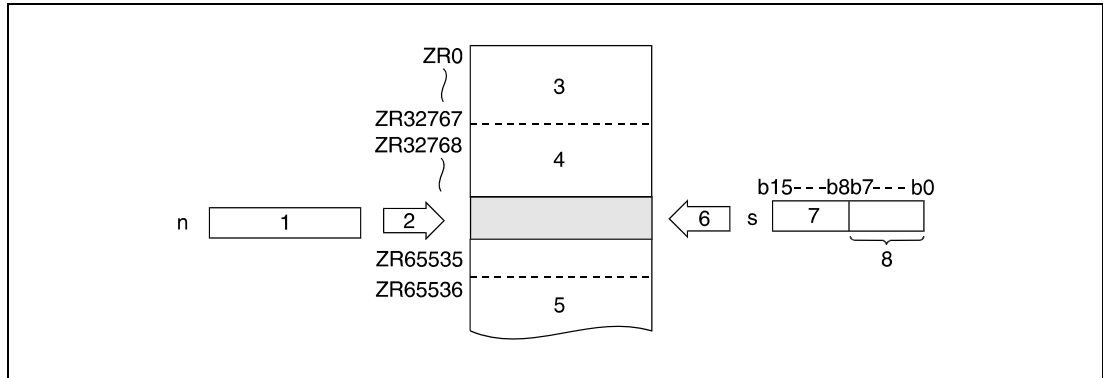
Операнд	Значение	Тип данных
n	Последовательный адрес байта, записываемый в регистр файлов.	BIN, 32 бита
s	Операнд, в котором сохраняются записываемые данные.	BIN, 16 бит

**Принцип действия**

**Непосредственная запись одного байта в регистр файлов**

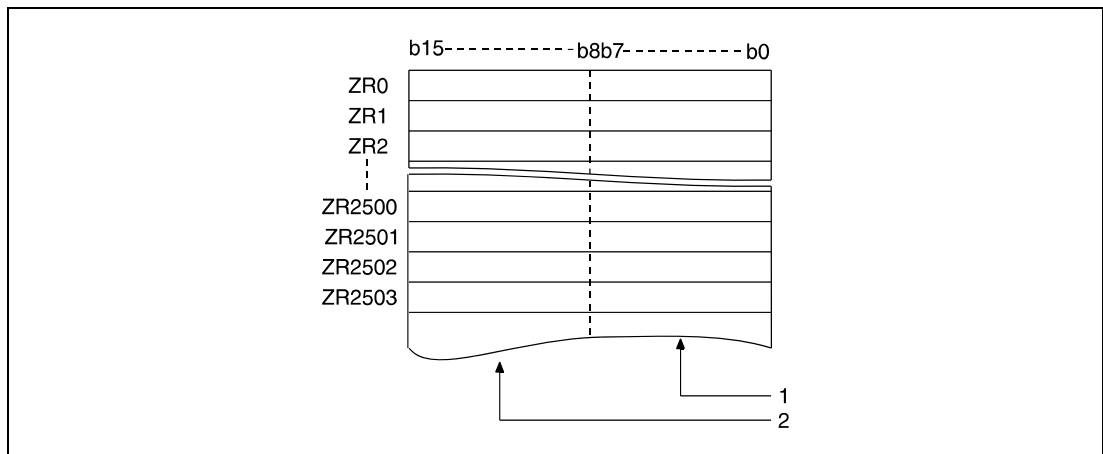
**ZRWRB Запись одного байта**

Команда ZRRDB записывает содержимое младшего байта операнда, указанного в s, в последовательный адрес байта регистра файлов, указанный в n. При адресации целевого байта блок регистров файлов не указывается. Старший байт операнда, указанного в s, игнорируется.



- 1 последовательный адрес байта
- 2 адресация
- 3 область регистра файлов для блока 0
- 4 область регистра файлов для блока 1
- 5 область регистра файлов для блока 2
- 6 запись данных
- 7 Этот байт при обработке программы игнорируется.
- 8 записываемый байт

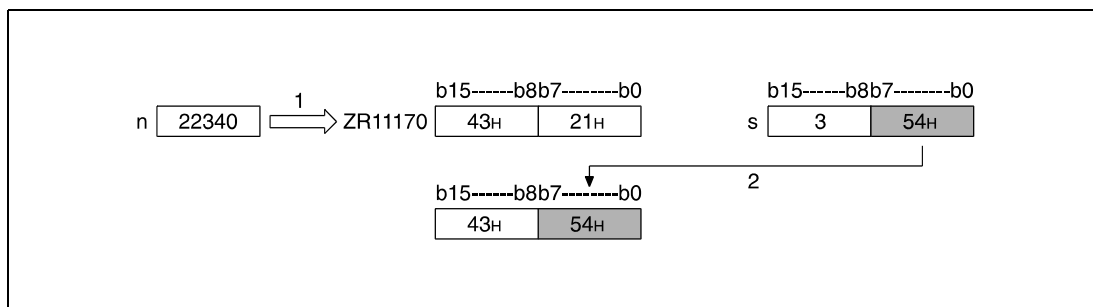
Взаимосвязь между адресами регистров файлов и соответствующими последовательными адресами байтов показана на следующей иллюстрации.



- 1 область памяти для четных адресов байтов (здесь: адреса от 0 до 5006)
- 2 область памяти для нечетных адресов байтов (здесь: адреса от 1 до 5007)

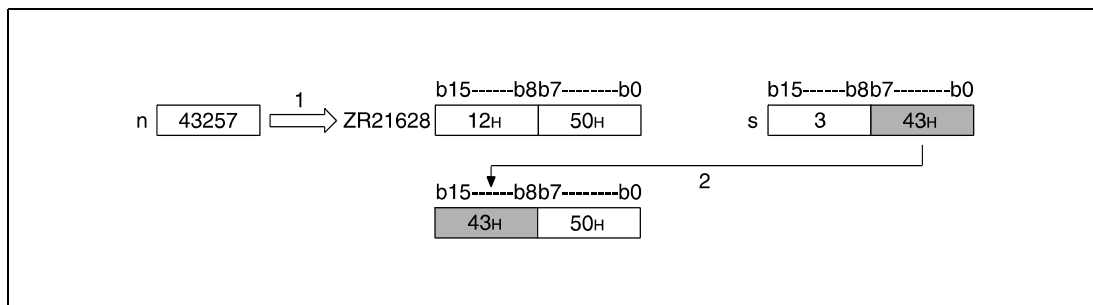


Например, если указан адрес байта 22340, то содержимое младшего байта операнда, указанного в s, записывается в младший байт регистра файлов ZR 11170.



- <sup>1</sup> адресация
- <sup>2</sup> запись байта
- <sup>3</sup> Этот байт при обработке программы игнорируется.

Если, например, указан адрес байта 43257, то содержимое младшего байта операнда, указанного в s, записывается в старший байт регистра файлов ZR21628.



- <sup>1</sup> адресация
- <sup>2</sup> запись байта
- <sup>3</sup> Этот байт при обработке программы игнорируется.

**Источники ошибок**

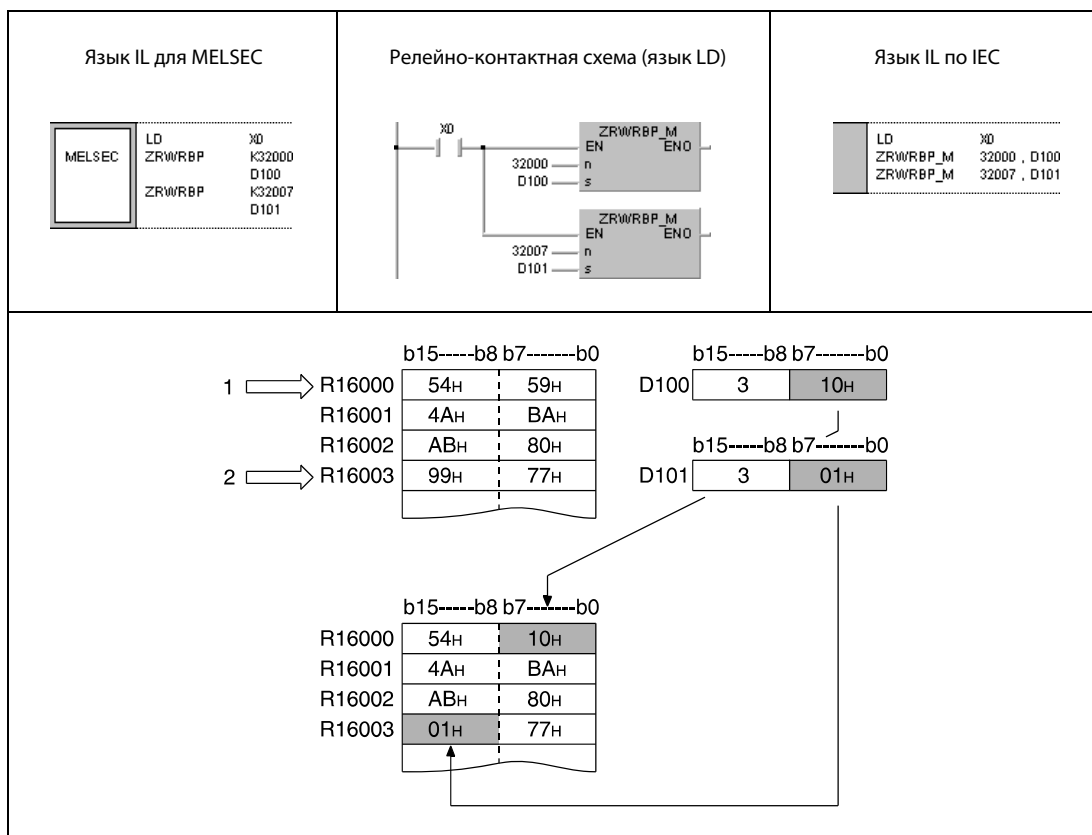
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанный в n адрес операнда (последовательный адрес байта) находится вне области, предусмотренной для сохранения (код ошибки 4101).

**Пример**

ZRWRBP

В следующей программе при положительном фронте сигнала X0 содержимое младших байтов регистров D100 и D101 записывается в младший байт регистра файлов R16000 (адрес байта 32000) и в старший байт регистра файлов R16003 (адрес байта 32007).



- <sup>1</sup> последовательный адрес байта 32000 (младший байт регистра файлов R16000)
- <sup>2</sup> последовательный адрес байта 32007 (старший байт регистра файлов R16003)
- <sup>3</sup> Эти байты при обработке игнорируются.

### 7.18.6 ADRSET, ADRSETP

Процессор

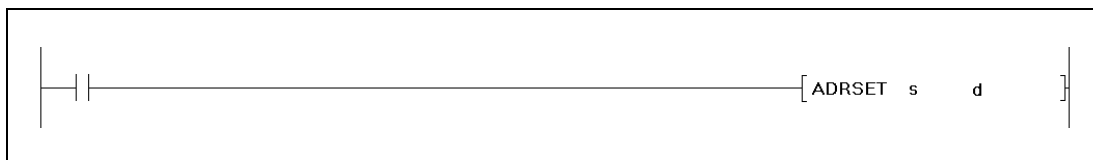
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	●	●	●	—	—	—	—	—	—	SM0	3
d	●	●	●	—	—	—	—	—	—		

**ПРИМЕЧАНИЕ** В GX IEC Developer эти команды использовать невозможно.

GX Developer



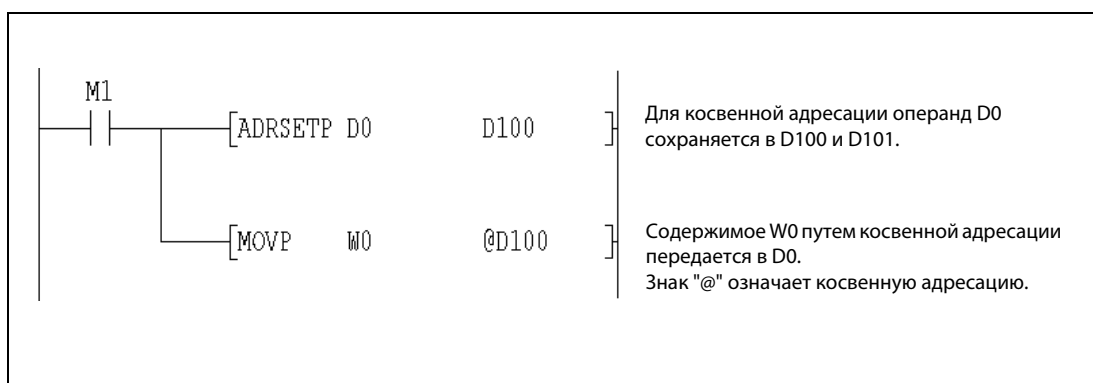
Переменные

Операнд	Значение	Тип данных
s	Операнд, к которому требуется обращаться путем косвенной адресации.	название операнда
d	Операнд, в котором сохраняется адрес косвенно адресуемого операнда.	BIN, 32 бита

Принцип действия

**Присвоение операнда для косвенной адресации**  
**ADRSET Сохранение адреса операнда**

Указанный в s операнд с целью косвенной адресации сохраняется в операнде d и d + 1. В s нельзя указывать битовый операнд.



Источники ошибок

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указан недопустимый операнд (код ошибки 4100).

### 7.18.7 KEY

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
		● <sup>1</sup>	● <sup>1</sup>	●	● <sup>2</sup>

<sup>1</sup> В случае процессоров AnA и AnU эту специализированную команду в стандартном редакторе МЭК (IEC) можно запрограммировать в качестве обычного элемента (функции), а в редакторе MELSEC – в сочетании с командами LEDA, C и R.

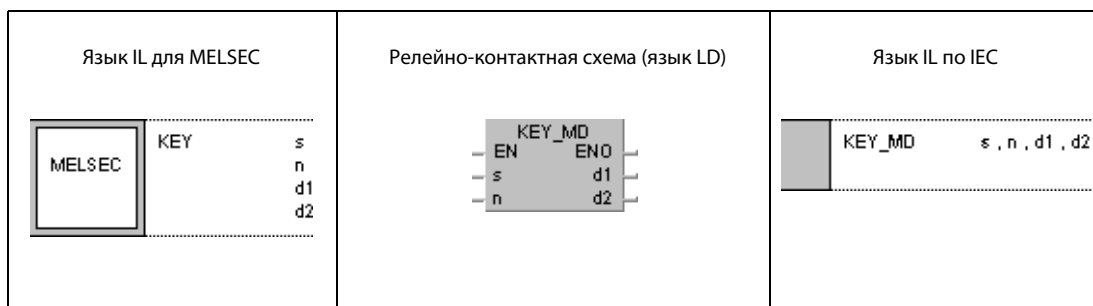
<sup>2</sup> кроме процессоров Q00J, Q00 и Q01

**Операнды MELSEC Q**

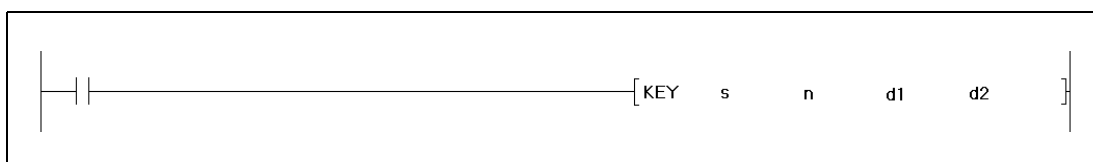
	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	● <sup>1</sup>	—	—	—	—	—	—	—	SM0	5	
n	●	●	●	●	●	●	●	—			
d1	—	●	●	—	—	—	—	—			
d2	●	●	●	●	●	●	—	—			

<sup>1</sup> только X

**GX IEC Developer**



**GX Developer**

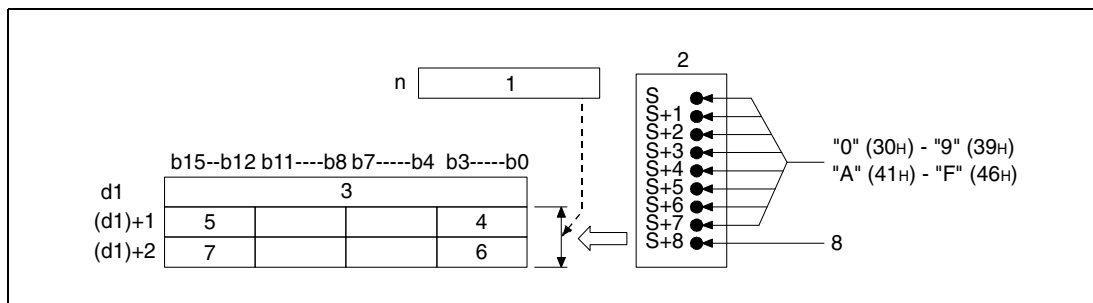


**Переменные**

Операнд	Значение	Тип данных	
		MELSEC	IEC
s	Первый адрес входных операндов (X), в которые записывается числовое значение.	бит	массив [1..9] данных типа BOOL
n	Количество вводимых числовых значений	BIN, 16 бит	ANY16
d1	Первый адрес операнда, в котором сохраняются введенные значения.	BIN, 16 бит	массив [1..3] данных типа ANY16
d2	Операнд, устанавливаемый по окончании ввода.	бит	BOOL

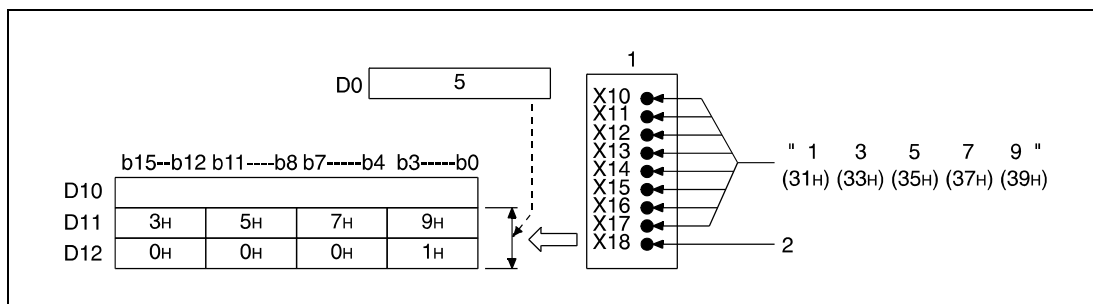
**Принцип действия** **Ввод числовых значений с клавиатуры**  
**KEY Команда ввода**

Команда KEY позволяет вводить с клавиатуры знаки ASCII от "0" (30н) до "9" (39н) и от "A" (41н) до "F" (46н) через входы (X), указанные в операндах с s + 0 (Array\_s[1]) по s + 7 (Array\_s[8]). Введенные через эти входы данные сохраняются в шестнадцатеричной кодировке в операндах, указанных с (d1) + 0 (Array\_d1[1]) по (d1) + 2 (Array\_d1[3]). Количество вводимых знаков указывается в n.



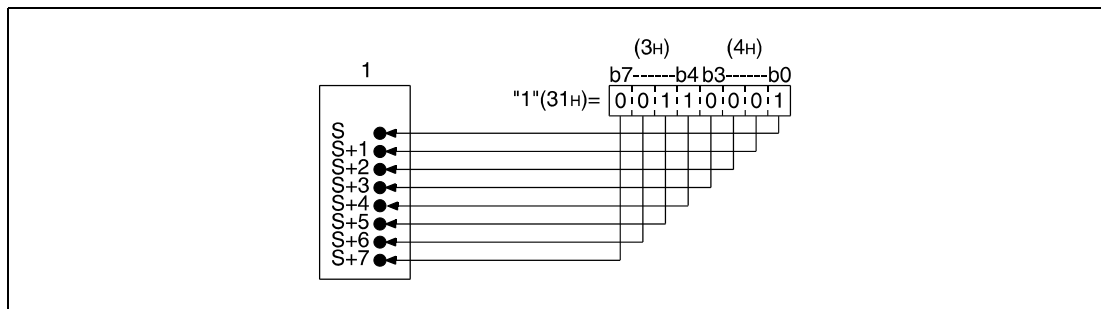
- 1 Количество знаков, которое должно быть введено.
- 2 входной модуль
- 3 количество введенных знаков
- 4 8-й введенный знак
- 5 5-й введенный знак
- 6 4-й введенный знак
- 7 1-й введенный знак
- 8 стробирующий сигнал

В примере на следующей иллюстрации в n указано 5, и значения от 1 (31н) до 5 (35н) вводятся через входы X10...X18 входного модуля.



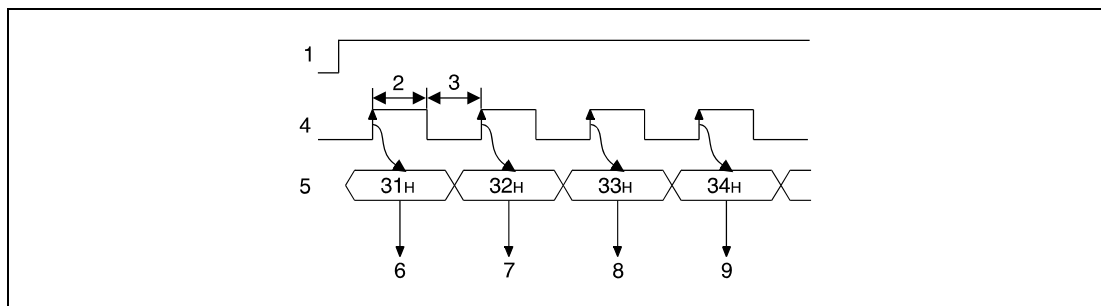
- 1 входной модуль
- 2 стробирующий сигнал

При вводе знаков ASCII через входы, указанные в операндах с  $s + 0$  (Array\_s[1]) по  $s + 7$  (Array\_s[8]) (X), знаки кодируются в 8-битной двоичной кодировке, как это показано ниже.



<sup>1</sup> входной модуль

После ввода знака ASCII через входы с  $s + 0$  (Array\_s[1]) по  $s + 7$  (Array\_s[8]) устанавливается stroбирующий сигнал ( $s + 8$ , Array\_s[9]), чтобы данные можно было сохранить внутри процессора. Время, на которое устанавливается или сбрасывается stroбирующий сигнал, должно быть больше длительности программного цикла. Если это время короче программного цикла, безупречное сохранение введенных данных не обеспечивается.



<sup>1</sup> условие выполнения для команды KEY

<sup>2</sup> Включен более одного цикла.

<sup>3</sup> Выключен более одного цикла.

<sup>4</sup> stroбирующий сигнал ( $s + 8$ , Array\_s[9])

<sup>5</sup> введенные данные в кодировке ASCII (от  $s + 0$  до  $s + 7$ , от Array\_s[1] до Array\_s[8])

<sup>6</sup> считывание знака "1"

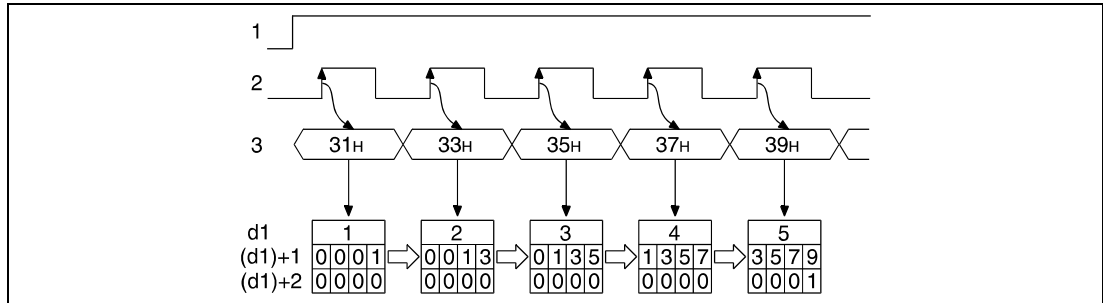
<sup>7</sup> считывание знака "2"

<sup>8</sup> считывание знака "3"

<sup>9</sup> считывание знака "4"

Выполнение команды KEY возможно только при установленном условии выполнения. Условие выполнения должно оставаться установленным до тех пор, пока не будет завершен ввод указанного в  $n$  количества знаков.

Количество введенных знаков сохраняется в (d1) + 0 (Array\_d[1]). Сами введенные знаки сохраняются в операндах, указанных в (d1) + 1 (Array\_d[2]) и (d1) + 2 (Array\_d[3]), в виде шестнадцатеричных кодов ASCII, образованных двоичными данными, т. е. на каждый знак выделяется по 4 бита. Шестнадцатеричные знаки от 0н до Fн отображаются двоичными значениями от "0000" до "1111".

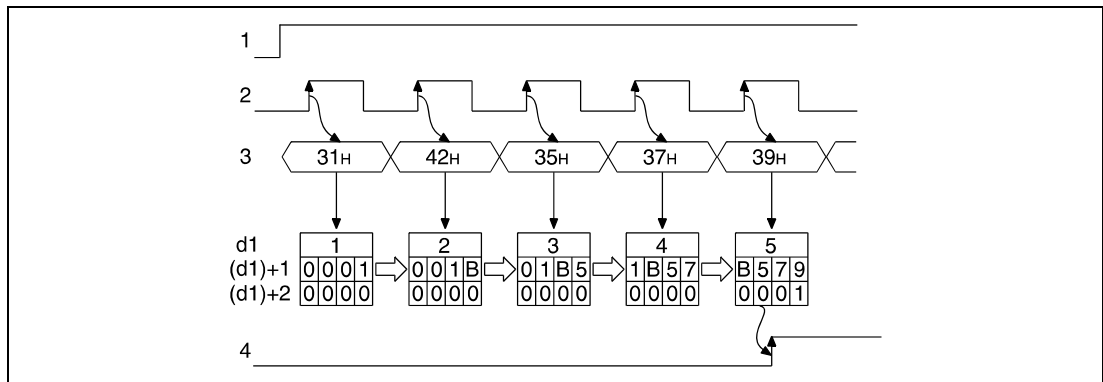


- 1 условие выполнения команды KEY
- 2 стробирующий сигнал (s + 8, Array\_s[9])
- 3 введенные данные в кодировке ASCII (от s + 0 до s + 7, от Array\_s[1] до Array\_s[8])

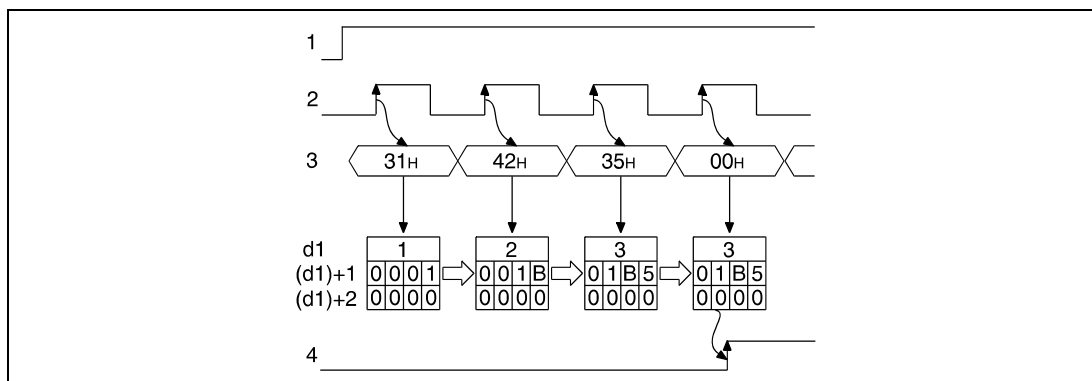
Количество вводимых знаков, указываемое в n, должно быть в диапазоне от 1 до 8.

Внутреннее сохранение введенных данных считается завершенным (и в связи с этим устанавливается операнд, указанный в d2), если введено количество знаков, указанное в n, или среди введенных знаков поступил код "00н". Это наглядно пояснено на следующих иллюстрациях. В этом примере в n указано 5.

На следующей иллюстрации ввод завершен после ввода 5 знаков. На второй иллюстрации ввод завершается после поступления кода "00н".



- 1 условие выполнения команды KEY
- 2 стробирующий сигнал (s + 8, Array\_s[9])
- 3 введенные данные в кодировке ASCII (от s + 0 до s + 7, от Array\_s[1] до Array\_s[8])
- 4 Ввод знаков завершен (устанавливается операнд, указанный в d2).



- <sup>1</sup> условие выполнения команды KEY  
<sup>2</sup> стробирующий сигнал ( $s + 8$ , Array\_s[9])  
<sup>3</sup> введенные данные в кодировке ASCII (от  $s + 0$  до  $s + 7$ , от Array\_s[1] до Array\_s[8])  
<sup>4</sup> Ввод знаков завершен (устанавливается операнд, указанный в d2).

Перед повторным вводом знаков необходимо стереть содержимое операндов, указанных с  $(d1) + 0$  (Array\_d1[1]) по  $(d1) + 2$  (Array\_d[3]), и сбросить операнд, указанный в d2. Если эти операнды не стерты и не сброшены, повторный ввод знаков не возможен.

#### Источники ошибок

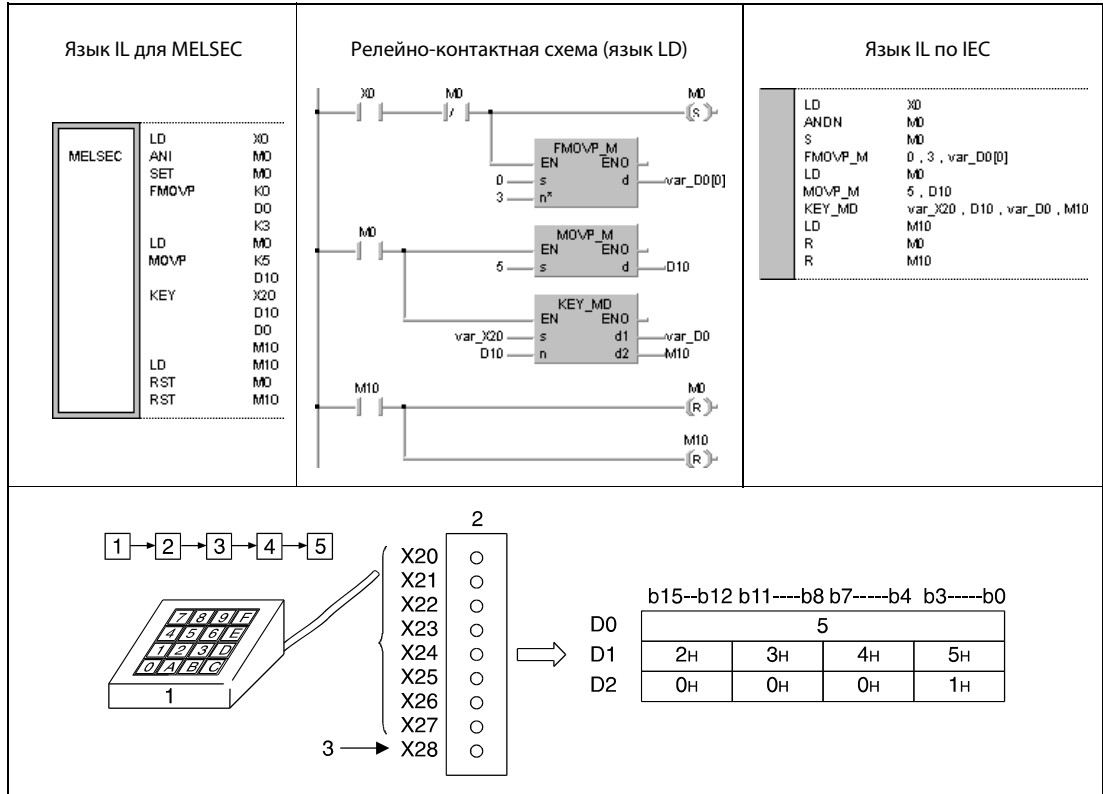
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанный в  $s$  операнд не является входом (X) (код ошибки 4100).
- Указанное в  $n$  количество вводимых знаков не находится в диапазоне между 1 и 8.



**Пример**

В следующей программе можно с клавиатуры ввести до 5 знаков через входы с X20 (var\_X20[0]) по X27 (var\_X20[7]). Эти знаки сохраняются в шестнадцатеричном формате в регистрах D1 (var\_D0[1]) и D2 (var\_D0[2]). Количество уже введенных знаков сохраняется в D0 (var\_D0[0]). Перед выполнением команды KEY стираются регистры с D0 (var\_D0[0]) по D2 (var\_D0[2]) и указывается количество вводимых данных 5. После выполнения команды KEY сбрасывается маркер M10 (завершение ввода). Стробящий сигнал подан на вход X28 (var\_X20[8]).



### 7.18.8 ZPUSH, ZPUSHP, ZPOP, ZPOPP

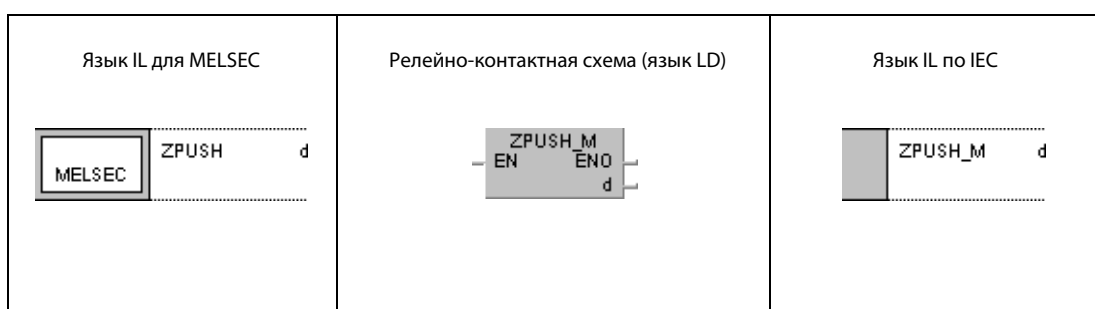
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

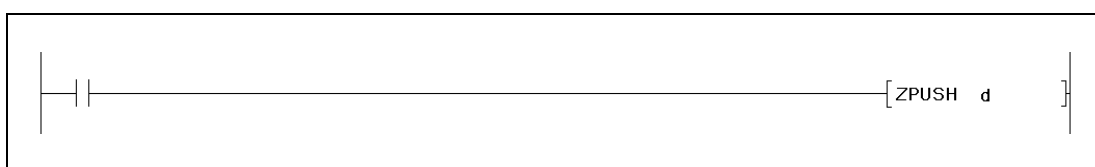
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
d	—	●	●	—	—	—	—	—	SM0	2	

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
d	Первый адрес операнда, в котором фиксируется содержимое индексных регистров.	BIN, 16 бит

**Принцип действия**

**Сохранение и восстановление содержимого индексных регистров**

**ZPUSH Сохранение содержимого индексных регистров**

Команда ZPUSH фиксирует содержимое индексных регистров с Z0 по Z15 в операндах, начиная с указанного в d.

Эти данные можно затем восстановить с помощью команды ZPOP. Эти команды могут быть размещены на различных уровнях вложений, имеющих в петле ZPUSH/ZPOP.

При применении этих команд на различных уровнях вложений необходимо для каждого выполнения команды ZPUSH помещать 18 регистров по 16 битов в операнды, начиная с указанного в d. Поэтому по адресу, начиная с указанного d, для выполнения команды ZPUSH должно иметься соответствующее место в памяти.

Распределение памяти, начиная с d, проиллюстрировано на следующем рисунке.



**ZPOP Восстановление содержимого индексных регистров**

Команда ZPOP восстанавливает содержимое индексных регистров, сохраненное с помощью команды ZPUSH. При этом данные считываются из области памяти, начиная с d, и снова записываются в соответствующие индексные регистры.

**Источники ошибок**

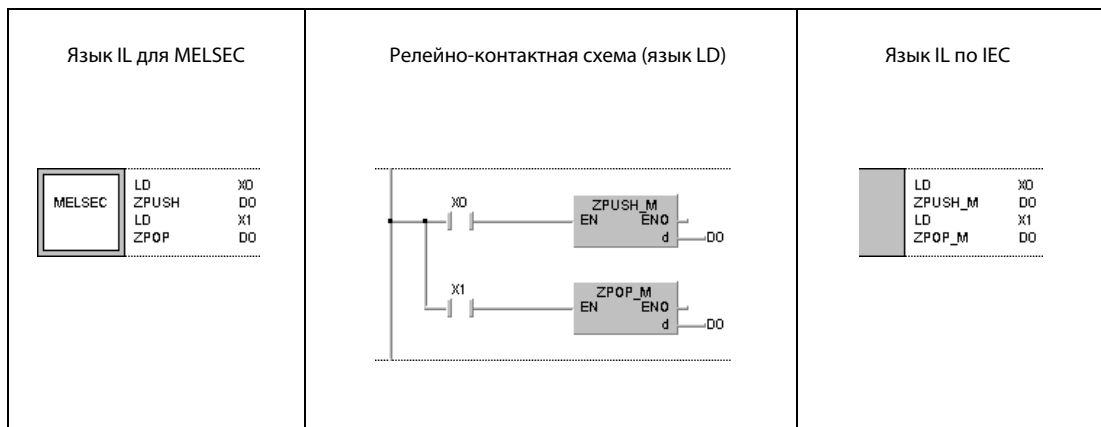
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Область памяти операндов, начиная с указанного в d, превышает область, предусмотренную для сохранения (код ошибки 4101).
- Содержимое операнда, указанного в d + 0, (количество сделанных сохранений) равно 0 (код ошибки 4100).

**Пример**

ZPUSH/ZPOP

В следующей программе по сигналу X0 индексные регистры сохраняются, а по команде X1 снова считываются.



### 7.18.9 EROMWR, EROMWRP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные U
	битовые	словные		битовые	словные						
s	—	●	—	—	—	—	—	—	SM0	5	
d1	—	—	●	—	—	—	—	—			
n	●	●	●	●	●	●	●	—			
d2	●	—	—	—	—	—	—	—			

**GX IEC Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Первый адрес операнда, в котором сохранены записываемые данные.	BIN, 16 бит
d1	Первый адрес регистров файлов EEPROM, в которые осуществляется запись.	
n	Количество записываемых слов данных	
d2	Операнд, устанавливаемый по окончании операции записи.	бит

<b>Принцип действия</b>	<b>Запись данных в регистр файлов EEPROM</b> <b>EROMWR/EROMWRP      Команда записи</b> <p>Команда EROMWR записывает n слов данных, хранящихся в s, в регистр файлов EEPROM, указанный в d1.</p> <p>По окончании операции записи устанавливается операнд, указанный в d2. После одного цикла этот операнд снова автоматически сбрасывается.</p> <p>Команда EROMWR выполняется до достижения команды END. Во время каждого прогона программы могут быть записаны 64 слова данных. Количество прогонов можно определить, разделив указанное в n количество слов данных на 64 и округлив результат вверх. Для определения времени обработки количество прогонов следует умножить на время прогона, приблизительно равное 10 мс.</p> <p>Указанные в s данные во время процесса записи нельзя обновлять, иначе некоторые данные могут быть утрачены.</p>
<b>Источники ошибок</b>	<p>В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:</p> <ul style="list-style-type: none"><li>● Область памяти указанного в n количества слов данных находится вне области операндов, указанных в s или d1 (код ошибки 4101).</li><li>● Указанный в d1 регистр файлов не существует или не является регистром файлов EEPROM (код ошибки 4101).</li></ul>

## 8 Команды коммуникации

### 8.1 Основные сведения

Процессоры серий QnA(S) можно использовать в системах сетевой коммуникации MELSECNET(II)/V/10. Процессоры серии "System Q" поддерживает MELSECNET/10 и MELSECNET/H.

**ПРИМЕЧАНИЯ** *Под используемыми в тексте обозначениями MELSECNET/10 и MELSECNET/H подразумеваются сети MELSECNET/10 и MELSECNET/H.*

*Под обозначением MELSECNET подразумеваются системы сетевой коммуникации MELSECNET(I), MELSECNET(II) и MELSECNET/B.*

С помощью команд коммуникации центральный процессор считывает в главную станцию ("хост-станцию") данные из других станций, подключенных к сети MELSECNET, MELSECNET/10 или MELSECNET/H.

### 8.2 Типы команд

Команды коммуникации подразделяются на следующие четыре группы:

1. Команды обновления данных

Эти команды обновляют данные в указанных модулях сетевой коммуникации.

2. Специализированные команды коммуникации

Это новые команды коммуникации, используемые в процессорах серий QnA и "System Q". Для коммуникации можно использовать несколько каналов модуля сетевой коммуникации.

3. Команды коммуникации, совместимые с серией "A"

Эти команды совпадают со специализированными командами коммуникации процессоров серии "A".

4. Считывание и запись информации маршрутизации

Эти команды считывают параметры маршрутизации из релейных станций и станций маршрутизации и записывают параметры в эти станции.

Для систем MELSECNET и MELSECNET/10 можно использовать только определенные команды коммуникации. Кроме того, возможные команды для MELSECNET/10 зависят от того, что из себя представляет целевая станция – модуль центрального процессора серии "A" или QnA, модуль центрального процессора "System Q" или удаленная станция ввода-вывода.

Обзор команд коммуникации дан в следующей таблице.

Тип	Значение
Команды обновления сетевых данных	Команды для обновления данных в модулях сетевой коммуникации
Специализированные команды коммуникации	Считывание данных центрального процессора из целевых станций и запись данных в целевые станции целевых сетей, передача данных на модули сетевой коммуникации в целевых станциях целевых сетей, считывание данных центрального процессора, переданных с помощью команды SEND, запрос данных от других станций (операции записи и считывания данных времени, операция RUN/STOP), считывание данных из специальных модулей и их запись в специальные модули в удаленных станциях ввода-вывода.
Команды коммуникации, совместимые с серией "A"	Считывание данных центрального процессора из целевых станций и их запись в целевые станции целевых сетей, считывание данных из локальных станций и их запись в локальные станции (эти операции возможны только на мастер-станциях), считывание данных из специальных модулей и их запись в специальные модули в удаленных станциях ввода-вывода.
Считывание и запись информации маршрутизации	Считывание и запись параметров маршрутизации (номер сети, номер релейной станции, номер станции маршрутизации).



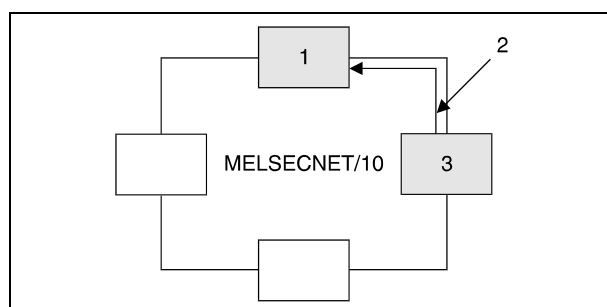
## 8.3 Области записи и считывания данных

### 8.3.1 MELSECNET/10

В MELSECNET/10 хост-станция способна выполнять операции записи и считывания в отношении станций в собственной сети, а при соответствующей адресации (параметрах маршрутизации) – и в отношении станций других сетей.

#### Операции записи и считывания в отношении станций собственной сети

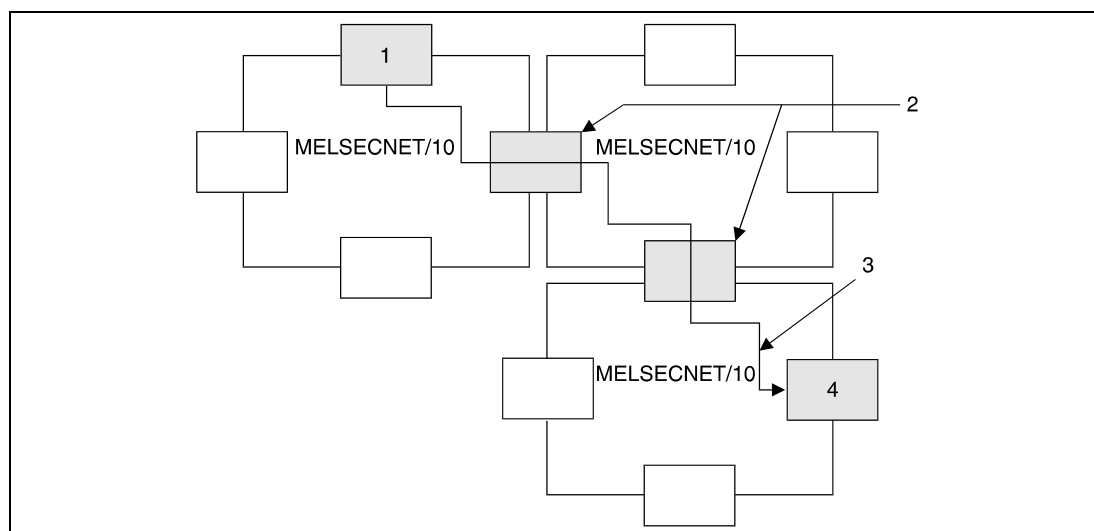
Для выполнения операций записи и считывания в отношении станций собственной сети номер сети целевой станции должен совпадать с номером сети модуля сетевой коммуникации хост-станции. Эта функция применяется для считывания и записи данных в станции одной сети.



- 1 Станция, выполняющая команду.
- 2 операция считывания
- 3 целевая станция

#### Операции записи и считывания в отношении станций другой сети

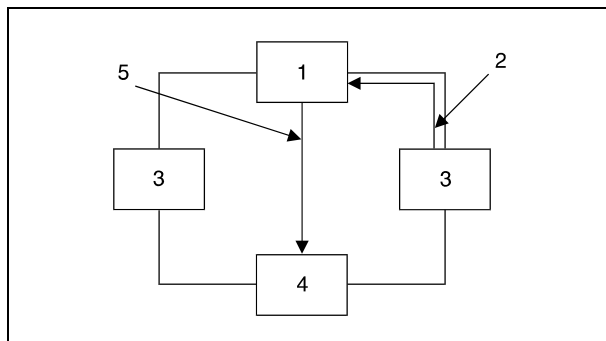
Для выполнения операций записи и считывания в отношении станций в других сетях номер сети целевой станции должен отличаться от номера сети модуля сетевой коммуникации хост-станции. Станция в сети хост-станции выполняет функцию релейной станции, перенаправляющей операции записи и считывания к целевой станции в другой сети.



- 1 Станция, выполняющая команду.
- 2 релейные станции (необходима настройка параметров маршрутизации)
- 3 операция считывания
- 4 целевая станция

### 8.3.2 MELSECNET

В сети MELSECNET(I/II/B) мастер-станция может выполнять операции записи и считывания по отношению к локальным станциям и удаленным станциям ввода-вывода.



- 1 мастер-станция
- 2 операция записи/считывания
- 3 локальная станция
- 4 удаленная станция ввода-вывода
- 5 операция записи/считывания в отношении специальных модулей

## 8.4 Специализированные команды коммуникации

Ниже разъяснены некоторые аспекты применения специализированных команд коммуникации для центральных процессоров серий QnA и "System Q".

### 8.4.1 Одновременное выполнение

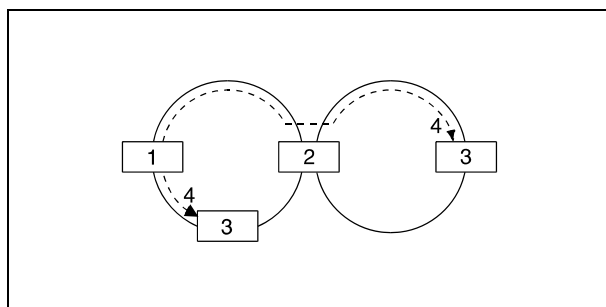
Модули сетевой коммуникации для системы MELSECNET/10 имеют 8 областей коммуникации, которые могут использоваться командами коммуникации. При использовании этих модулей одновременное выполнение нескольких команд коммуникации в одной области коммуникации не возможно. Если в одной области коммуникации центрального процессора требуется выполнять несколько команд коммуникации, следует обеспечить поочередную обработку этих команд путем блокировки с помощью операнда, устанавливаемого после полного выполнения команды записи/считывания.

### 8.4.2 Конец передачи данных

При использовании специализированных команд коммуникации можно выбрать, должно ли подтверждаться окончание обработки или передачи данных.

#### Подтверждение конца передачи данных

На рисунке ниже изображен режим, при котором конец обработки или передачи данных подтверждается – после того, как данные были записаны в адресованный канал целевой станции (при операциях считывания можно выбрать только этот режим).



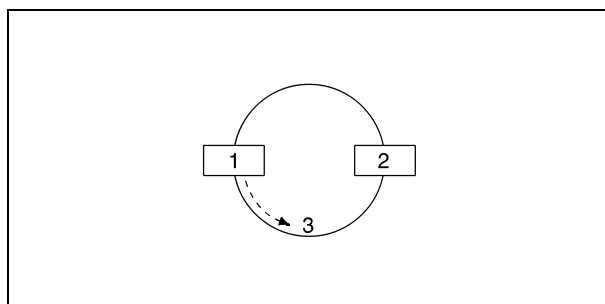
- 1 источник выполнения
- 2 релейная станция
- 3 целевая станция
- 4 конец обработки/передачи

**Без подтверждения конца передачи данных**

На следующих иллюстрациях показаны режимы, при которых конец обработки или передачи не подтверждается.

В пределах одной сети:

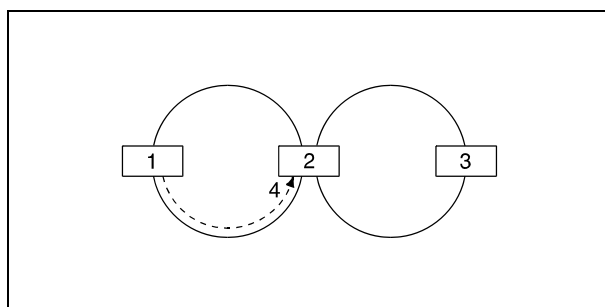
Считается, что конец обработки или передачи достигнут, если хост-станция передала все данные.



- 1 источник выполнения
- 2 целевая станция
- 3 конец обработки/передачи

Между различными сетями:

Считается, что конец обработки или передачи достигнут, если переданные данные достигли релейной станции в главной сети (хост).



- 1 источник выполнения
- 2 релейная станция
- 3 целевая станция
- 4 конец обработки/передачи

**ПРИМЕЧАНИЯ** Для обеспечения правильной обработки данных рекомендуется использовать режим с подтверждением конца передачи данных.

Если указан режим без подтверждения конца передачи данных, то после передачи данных передающая станция извещается о конце передачи независимо от того, имелись ли ошибки во время передачи. Кроме того, если команды коммуникации поступают от нескольких станций, целевая станция получает сообщение об ошибке "Входная буферная память полна". Это происходит даже в том случае, если данные были переданы правильно. Однако передающая станция в ответ на это сообщение прекращает передачу данных.

## 8.5 Команды обновления данных

С помощью следующих команд необходимо обновлять данные в модулях сетевой коммуникации. Таблица содержит обзор команд.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе IEC	Целевая станция в MELSECNET/10			MELSECNET
			Центр. проц. QnA, "System Q"	Центр. проц. "A"	Удаленная станция ввода-вывода	
Команды обновления данных	ZCOM	ZCOM_J_M	—	—	—	—
		ZCOM_JP_M				
		ZCOM_U_M				
		ZCOM_UP_M				

### 8.5.1 ZCOM

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

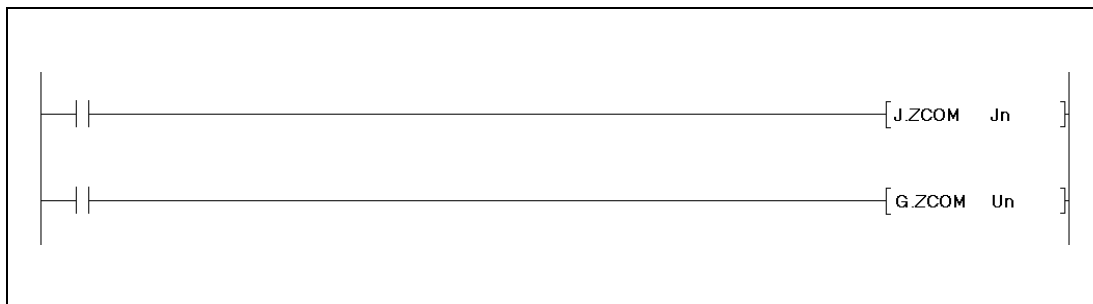
**Операнды MELSEC Q**

Операнды										Флаг ошибки	Шагов
Внутренние операнды (сист., польз.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные							
—	—	—	—	—	—	—	—	—	—	SM0	5

**GX IEC Developer**



**GX Developer (процессоры QnA)**



**GX Developer (System Q)**



**Переменные**

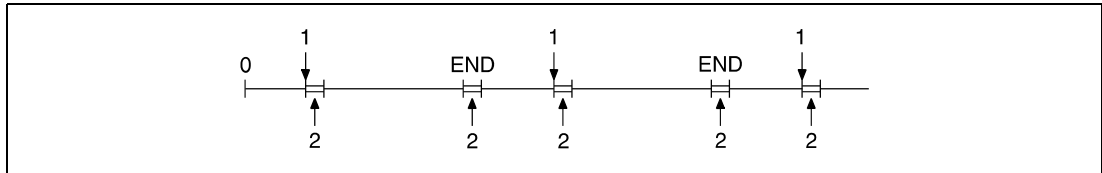
Операнд	Значение	Тип данных
Jn	Номер сети хост-станции	BIN, 16 бит
Un	Головной адрес ввода-вывода сети хост-станции	

**Принцип действия**

**Обновление сетевых данных**

**ZCOM Обновление данных в модулях сетевой коммуникации**

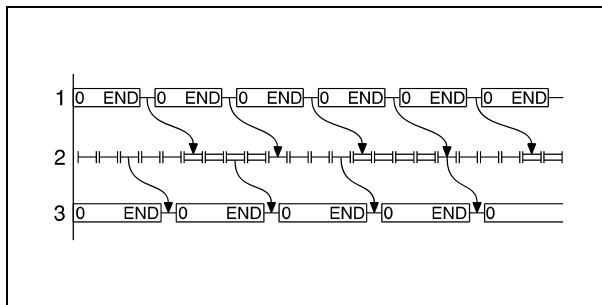
При выполнении команды ZCOM центральный процессор прерывает выполнение основной программы и обновляет данные в модулях сетевой коммуникации, указанных в Jn и Un.



- 1 выполнение команды ZCOM
- 2 обновление данных

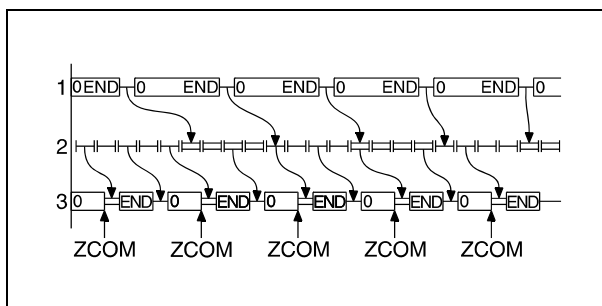
Если время цикла основной программы хост-станции превышает время цикла других станций, применение команды ZCOM обеспечивает правильную регистрацию данных из других станций.

На рисунке ниже показан пример обмена данными без применения команды ZCOM.



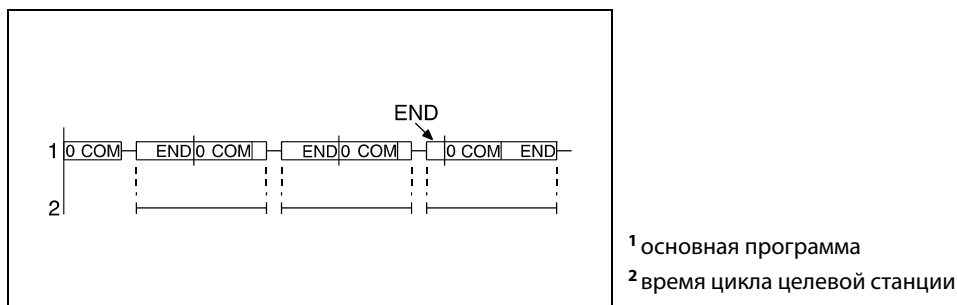
- 1 программа управляющей станции
- 2 цикл целевой станции
- 3 программа обычной станции

На рисунке ниже показан обмен данными с применением команды ZCOM.



- 1 программа управляющей станции
- 2 цикл целевой станции
- 3 программа обычной станции

Если время цикла целевой станции превышает время цикла основной программы, применение команды ZCOM не улучшает обмен данными.



Команду ZCOM можно применять в основной программе сколь угодно часто. Однако следует учитывать, что при каждом обновлении данных время цикла основной программы увеличивается на время выполнения обновления.

Команда ZCOM не может применяться при следующих операциях:

- коммуникация между центральным процессором и периферийными устройствами
- контроль других станций
- считывание буферной памяти из других специальных модулей через модуль Computer-Link.

**ПРИМЕЧАНИЕ** Указав *Un*, можно обращаться не только к модулям сетевой коммуникации, но и к специальным модулям. Так как в этом случае происходит обновление буферной памяти специального модуля, эту команду можно использовать вместо команд FROM и TO.

**Источники ошибок** В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

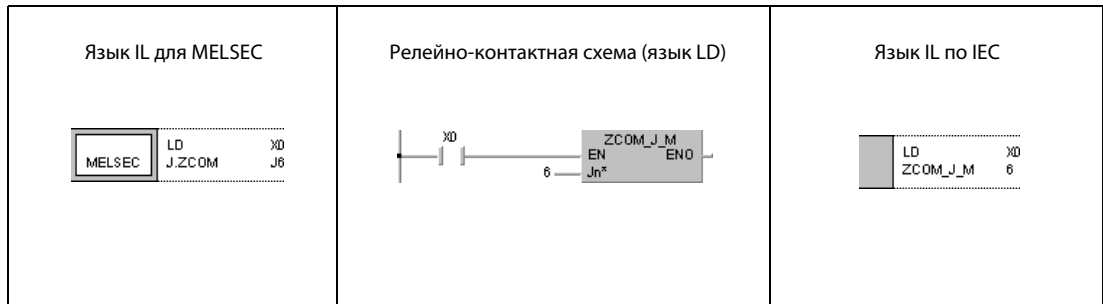
- Сеть с указанным номером не соединена с хост-станцией (код ошибки 4102).
- Модуль, находящийся по указанному адресу ввода-вывода, не является модулем сетевой коммуникации или связи (код ошибки 2111).

**ПРИМЕЧАНИЯ** Для исключительной обработки общих данных следует применять команду COM.

Необходимо учитывать, что могут возникать противоречивые данные. Например, какой-нибудь операнд может изменяться за время одного программного цикла.

**Пример 1** J.ZCOM

Следующая программа при включенном входе X0 обновляет данные в модуле сетевой коммуникации с номером сети 6.



**Пример 2** G.ZCOM

Следующая программа при включенном входе X0 обновляет данные в модуле сетевой коммуникации, находящемся на адресах ввода-вывода с X/Y30 по X/Y4F.





## 8.6 Специализированные команды коммуникации для серии QnA

Эти команды позволяют обмениваться данными между станциями, оснащенными центральным процессором QnA, а также между центральными процессорами QnA и удаленными станциями ввода-вывода в сети MELSECNET/10. Следующая таблица содержит обзор команд.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе IEC	Целевая станция в MELSECNET/10			MELSECNET
			центр. проц. QnA	центр. проц. "A"	удаленная станция ввода-вывода	
Считывание данных центрального процессора QnA из целевых станций целевых сетей	READ	READ_M	●	—	—	—
		READP_M				
		READ_JP_M				
		READ_UP_M				
	SREAD	SREAD_JP_M	●	—	—	—
SREAD_UP_M						
Запись данных центрального процессора QnA в целевые станции целевых сетей	WRITE	WRITE_JP_M	●	—	—	—
		WRITE_UP_M				
	SWRITE	SWRITE_M	●	—	—	—
		SWRITE_JP_M				
		SWRITE_UP_M				
Передача данных на модули сетевой коммуникации в целевых станциях целевых сетей	SEND	SEND_M	●	—	—	—
		SEND_4_M				
		SEND_4_P_M				
		SEND_JP_M				
		SEND_UP_M				
Считывание данных центрального процессора QnA, переданных с помощью команды SEND	RECV	RECV_M	●	—	—	—
		RECV_P_M				
		RECV_JP_M				
		RECV_UP_M				
Запрос данных от других станций (операции записи и считывания данных времени, запрос режима RUN/STOP)	REQ	REQ_M	●	—	—	—
		REQ_P_M				
		REQ_JP_M				
		REQ_UP_M				
Считывание данных из специальных модулей в удаленных станциях ввода-вывода	ZNFR	ZNFR_JP_M	—	—	●	—
		ZNFR_UP_M				
Запись данных в специальные модули в удаленных станциях ввода-вывода	ZNT0	ZNT0_J_M	—	—	●	—
		ZNT0_U_M				
		ZNT0_JP_M				
		ZNT0_UP_M				

8.6.1 READ

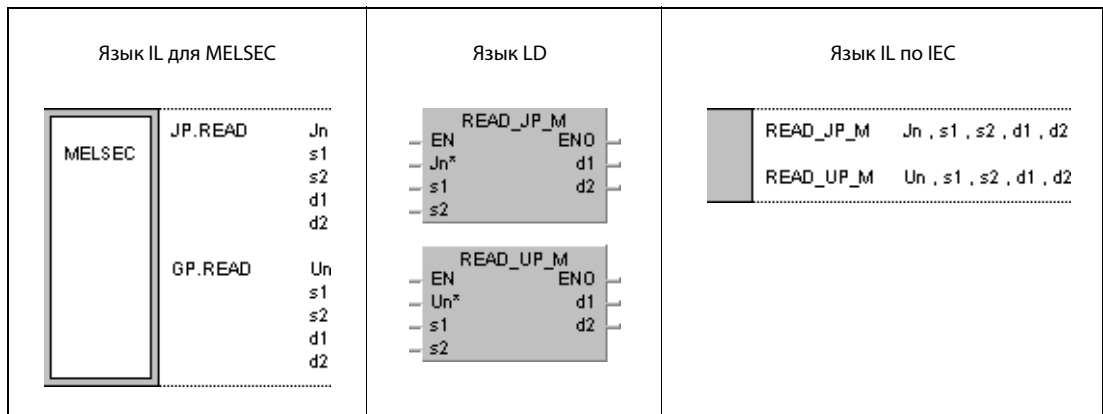
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (сист., польз.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	●	—	SM0	9
s2	—	●	●	—	—	—	—	●	—		
d1	—	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		

GX IEC Developer



Переменные

Операнд	Значение	Тип данных	
		MELSEC	IEC
Jn	Номер сети хост-станции ● <sup>1</sup>		
Un	Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции ● <sup>2</sup>	BIN, 16 бит	ANY16
s1	Первый операнд хост-станции, в котором сохранены управляющие данные.	адрес	массив [1..18] данных типа ANY16
s2	Первый операнд станции, в котором сохранены считываемые данные.		ANY16
d1	Первый операнд хост-станции, в котором сохраняются считанные данные.		
d2	Операнд, устанавливаемый после полной обработки команды на один программный цикл.	бит	BOOL

**ПРИМЕЧАНИЯ**

- <sup>1</sup> Номер сети хост-станции должен находиться в диапазоне от 1 до 239.  
Сеть с номером 254 с помощью настроек конфигурируется так, чтобы другие станции могли обращаться к активной станции.
- <sup>2</sup> Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции должен находиться в диапазоне от 0 до FEn. Следует иметь в виду, что в Un компилятор ожидает шестнадцатеричного указания адреса.  
Десятичный адрес автоматически преобразуется в шестнадцатеричное число.

Команда READ может быть выполнена только в том случае, если целевая станция построена на процессоре QnA.

При использовании процессора серии "A" в сети MELSECNET/10 команду READ не следует применять.

В качестве номера целевой станции можно указывать только номера станций с процессорами QnA.

**Обзор операндов управляющих данных**

Операнд	Значение	Функция	Диапазон	Устанавливает
(s1) + 0 Array_s1[1]	Режим выполнения	Подтверждение приема установлено: (бит 0 (b0) = 1, фиксированная настройка)	0001h 0081h	пользователь
	Режим завершения обработки ошибки	Сохранение момента времени, в который была завершена обработка ошибки: – без сохранения данных времени, бит 7 (b7) = 0 – сохранение данных времени, бит 7 (b7) = 1 (данные времени, начиная с (s1) + 11 (Array_s1[12]))		
(s1) + 1 Array_s1[2]	Состояние после завершения обработки команды	Индикация состояния по окончании обработки команды: 0 = ошибок нет (нормальное завершение обработки) не равно 0: = код ошибки ● <sup>3</sup>	—	система
(s1) + 2 Array_s1[3]	Канал, используемый хост-станцией	Указание канала, который будет применять хост-станция.	от 1 до 8	пользователь
(s1) + 3 Array_s1[4]	Фиктивный операнд	Не используется.	0	—
(s1) + 4 Array_s1[5]	Номер сети целевой станции	Номер сети станции, из которой происходит считывание.	от 1 до 239 254 ● <sup>4</sup>	пользователь
(s1) + 5 Array_s1[6]	Номер целевой станции	Указание номера целевой станции.	от 1 до 64	пользователь
(s1) + 6 Array_s1[7]	Фиктивный операнд	Не используется.	—	—
(s1) + 7 Array_s1[8]	Количество попыток передачи	Указание количества попыток, которое требуется предпринять, чтобы полностью обработать команду READ за контрольное время, заданное в (s1) + 8 (Array_s1[9]).	от 1 до 15	пользователь
	Количество выполненных попыток передачи	Сохранение выполненных попыток передачи.		
(s1) + 8 Array_s1[9]	Настройка сторожевого таймера для контроля времени приема	Указание контрольного времени для операций считывания READ в секундах. Если операция не завершается за указанное время, передача повторяется количество раз, указанное в (s1) + 7 (Array_s1[8]).	от 1 до 32767 0 = 10 с (фиксированная настройка)	пользователь
(s1) + 9 Array_s1[10]	Длина принимаемых данных	Количество считываемых блоков данных	от 1 до 480	пользователь
(s1) + 10 Array_s1[11]	Фиктивный операнд	Не используется.	—	—
(s1) + 11 Array_s1[12]	Флаг сохранения данных времени (устанавливается только при наличии ошибки)	Этот флаг сохраняет состояние опции сохранения данных времени, указанной в (s1) + 0 (Array_s1[1]): – без сохранения данных времени = 0 – сохранение данных времени = 1	—	система
(s1) + 12 Array_s1[13]	Данные времени устанавливаются только при наличии ошибки)	старший байт = год (от 0 до 99) младший байт = месяц (от 1 до 12)	—	система
(s1) + 13 Array_s1[14]		старший байт = день (от 1 до 31) младший байт = час (от 0 до 23)		
(s1) + 14 Array_s1[15]		старший байт = минута (от 0 до 59) младший байт = секунда (от 0 до 59)		
(s1) + 15 Array_s1[16]		старший байт = 00h младший байт = день недели (от 0 до 6) (воскресенье = 0, суббота = 6)		
(s1) + 16 Array_s1[17]	Номер сети, в которой возникла ошибка	Сохранение номера сети станции, в которой возникла ошибка. Номер сети находится в диапазоне от 1 до 239.	—	система

Операнд	Значение	Функция	Диапазон	Устанавливает
(s1) + 17 Array_s1[18]	Номер станции, в которой возникла ошибка	Сохранение номера станции, в которой возникла ошибка. Сохранение не выполняется, если состояние после завершения выполнения команды гласит "Канал используется". Номер станции находится в диапазоне от 1 до 64.	—	система

- <sup>3</sup> Дополнительная информация о кодах ошибок имеется в руководстве "MELSECNET/10 для систем сетевой коммуникации QnA".
- <sup>4</sup> Если в Jn указано значение 254, выбран номер сети 254.

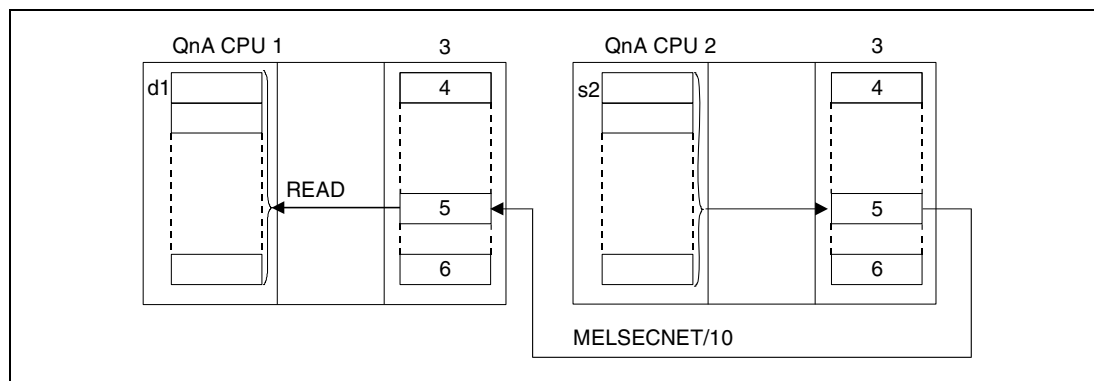
**Принцип действия**

**Считывание данных из словных операндов других станций**

**READ Команда считывания**

Команда READ считывает данные, хранящиеся начиная с s2 в станции, подключенной к сети MELSECNET/10. Номер станции и номер сети указаны в управляющих данных. Считанные из станции данные сохраняются в хост-станции, начиная с d1.

По окончании операции считывания в целевой станции устанавливается операнд d2.



- 1 хост-станция
- 2 целевая станция
- 3 модуль сетевой коммуникации
- 4 канал 1
- 5 канал n
- 6 канал 8

С помощью релейной станции и установленных параметров маршрутизации можно обращаться и к станциям в других сетях.

Команды коммуникации не могут выполняться одновременно из нескольких источников с общим доступом к одним и тем же каналам. При одновременном выполнении команд из двух или более источников выполнение других команд коммуникации предотвращается с помощью режима квитирования (handshake) обеих активных станций.

Состояние выполнения и состояние после завершения (нормальное, ненормальное) команды READ можно проверить с помощью

- флага (●<sup>5</sup>) используемого коммуникационного канала,
- операнда, устанавливаемого по окончании операции считывания (d2), и
- индикации состояния после завершения операции (безошибочное или ошибочное завершение передачи) ((d2) + 1)

Проверка выполняется следующим образом:

Флаг коммуникационного канала

Этот флаг устанавливается во время выполнения команды READ. Флаг сбрасывается при обработке команды END того цикла, в котором была завершена операция считывания.

Операнд хост-станции, показывающий завершение операции считывания

Этот операнд устанавливается при обработке команды END того цикла, в котором завершена операция считывания. При обработке следующей команды END этот операнд снова сбрасывается.

Индикация состояния после завершения считывания

Этот операнд устанавливается в зависимости от результата после завершения операции считывания.

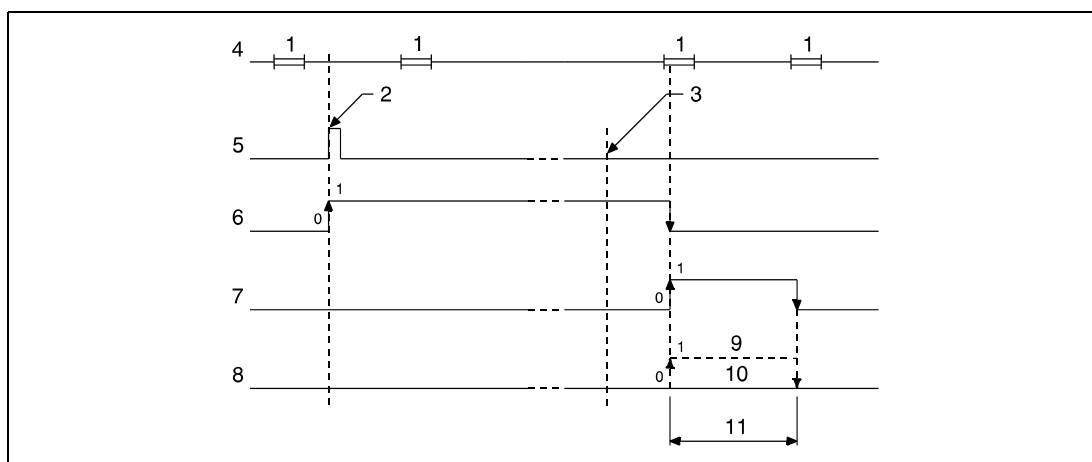
При завершении нормальной (безошибочной) передачи операнд остается сброшенным.

После завершения ненормальной передачи (т. е. передачи, содержащей ошибки) этот операнд устанавливается при обработке команды END того цикла, в котором была завершена обработка команды READ. При обработке следующей команды END этот операнд снова сбрасывается.

**ПРИМЕЧАНИЕ** ●<sup>5</sup> В следующей таблице указана взаимосвязь между номерами каналов и флагами коммуникационных каналов.

Номер канала	1	2	3	4	5	6	7	8
Флаг коммуникационного канала	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

На рисунке ниже изображены операции хост-станции во время выполнения команды READ.



- 1 обработка команды END
- 2 выполнение команды READ
- 3 завершение операции
- 4 программа хост-станции
- 5 команда READ
- 6 флаг коммуникационного канала
- 7 операнд хост-станции, устанавливаемый по окончании операции ((d2)
- 8 индикация состояния после завершения операции ((d2) + 1)
- 9 завершение ненормальной (ошибочной) передачи
- 10 завершение нормальной (безошибочной) передачи
- 11 один цикл

**Источники  
ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Содержимое управляющих данных находится вне допустимого диапазона (код ошибки 4100).
- Сеть с номером Jn не соединена со станцией (код ошибки 4102).
- Модуль по адресу ввода-вывода Un не является модулем сетевой коммуникации (код ошибки 2111).



### 8.6.2 SREAD

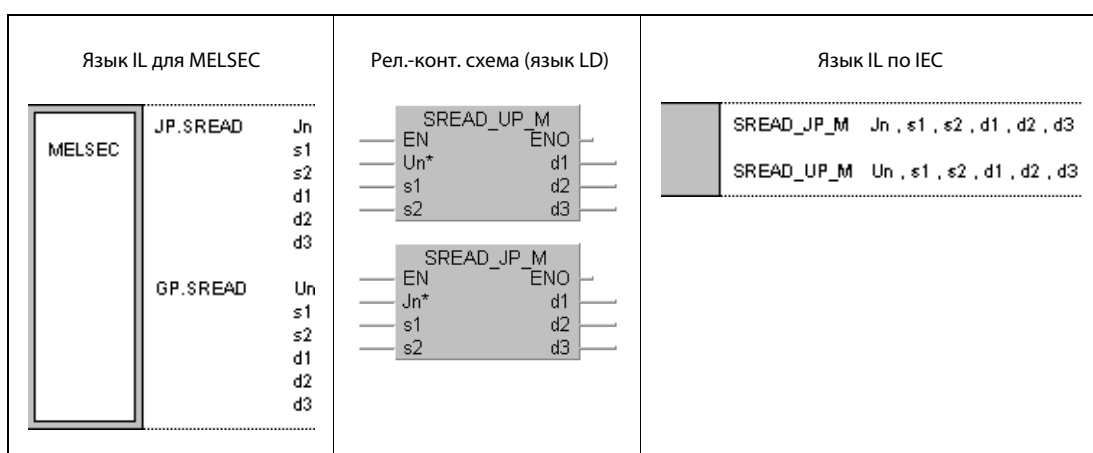
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (сист., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	●	—	SM0	10
s2	—	●	●	—	—	—	—	●	—		
d1	—	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		
d3	●	●	●	—	—	—	—	—	—		

GX IEC Developer



Переменные

Операнд	Значение	Тип данных	
		MELSEC	IEC
Jn	Номер сети хост-станции ● <sup>1</sup>	BIN, 16 бит	ANY16
Un	Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции ● <sup>2</sup>		
s1	Первый операнд хост-станции, в котором сохранены управляющие данные.	адрес	массив [1..18] типа ANY16
s2	Первый операнд станции, в котором сохранены считываемые данные.		
d1	Первый операнд хост-станции, в котором сохраняются считанные данные.		
d2	Операнд хост-станции, устанавливаемый после полной обработки команды на один программный цикл.	бит	BOOL
d3	Операнд целевой станции, устанавливаемый после полной обработки команды на один программный цикл.		

**ПРИМЕЧАНИЯ**

- <sup>1</sup> Номер сети хост-станции должен находиться в диапазоне от 1 до 239.  
*Сеть номер 254 с помощью настроек конфигурируется так, чтобы другие станции могли обращаться к активной станции.*
- <sup>2</sup> Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции должен находиться в диапазоне от 0 до FEn. Следует иметь в виду, что в Un компилятор ожидает шестнадцатеричного указания адреса. Десятичный адрес автоматически преобразуется в шестнадцатеричное число.

*Выполнение команды SREAD возможно только в том случае, если в целевой станции используется процессор серии QnA.*

*При использовании процессора серии "A" в сети MELSECNET/10 команду SREAD не следует применять.*

*В качестве номеров целевых станций можно указывать только номера станций с центральными процессорами серии QnA.*

**Обзор операндов управляющих данных**

Операнд	Значение	Функция	Диапазон	Устанавливает
(s1) + 0 Array_s1[1]	Режим выполнения	Подтверждение приема установлено: (бит 0 (b0) = 1, фиксированная настройка)	0001h 0081h	пользователь
	Режим завершения обработки ошибки	Сохранение момента времени, в который была завершена обработка ошибки: – без сохранения данных времени, бит 7 (b7) = 0 – сохранение данных времени, бит 7 (b7) = 1 (данные времени начиная с (s1) + 11 (Array_s1[12]))		
(s1) + 1 Array_s1[2]	Состояние после завершения обработки команды	Индикация состояния по окончании обработки команды: – 0 = ошибок нет (нормальное завершение обработки); не равно 0: = код ошибки ● <sup>3</sup>	—	система
(s1) + 2 Array_s1[3]	Канал, используемый хост-станцией	Указание канала, который будет применять хост-станция.	от 1 до 8	пользователь
(s1) + 3 Array_s1[4]	Фиктивный операнд	Не используется.	0	—
(s1) + 4 Array_s1[5]	Номер сети целевой станции	Номер сети станции, из которой происходит считывание.	от 1 до 239 254 ● <sup>4</sup>	пользователь
(s1) + 5 Array_s1[6]	Номер целевой станции	Указание номера целевой станции.	от 1 до 64	пользователь
(s1) + 6 Array_s1[7]	Фиктивный операнд	Не используется.	—	—
(s1) + 7 Array_s1[8]	Количество попыток передачи	Указание количества попыток, которое требуется предпринять, чтобы полностью обработать команду SREAD за контрольное время, заданное в (s1) + 8 (Array_s1[9]).	от 1 до 15	пользователь
	Количество выполненных попыток передачи	Сохранение выполненных попыток передачи		
(s1) + 8 Array_s1[9]	Настройка сторожевого таймера для контроля времени приема	Указание контрольного времени для операций считывания SREAD в секундах. Если операция не завершается за указанное время, передача повторяется количество раз, указанное в (s1) + 7 (Array_s1[8]).	от 1 до 32767 0 = 10 с (фиксированная настройка)	пользователь
(s1) + 9 Array_s1[10]	Длина принимаемых данных	Количество считываемых блоков данных	от 1 до 480	пользователь
(s1) + 10 Array_s1[11]	Фиктивный операнд	Не используется.	—	—
(s1) + 11 Array_s1[12]	Флаг сохранения данных времени (устанавливается только при наличии ошибки)	Этот флаг сохраняет состояние опции сохранения данных времени, указанной в (s1) + 0 (Array_s1[1]): без сохранения данных времени = 0 сохранение данных времени = 1.	—	система
(s1) + 12 Array_s1[13]	Данные времени устанавливаются только при наличии ошибки)	старший байт = год (от 0 до 99) младший байт = месяц (от 1 до 12)	—	система
(s1) + 13 Array_s1[14]		старший байт = день (от 1 до 31) младший байт = час (от 0 до 23)		
(s1) + 14 Array_s1[15]		старший байт = минута (от 0 до 59) младший байт = секунда (от 0 до 59)		
(s1) + 15 Array_s1[16]		старший байт = 00h младший байт = день недели (от 0 до 6) (воскресенье = 0, суббота = 6)		
(s1) + 16 Array_s1[17]	Номер сети, в которой возникла ошибка	Сохранение номера сети станции, в которой возникла ошибка. Номер сети находится в диапазоне от 1 до 239.	—	система

Операнд	Значение	Функция	Диапазон	Устанавливает
(s1) + 17 Array_s1[18]	Номер станции, в которой возникла ошибка	Сохранение номера станции, в которой возникла ошибка. Сохранение не выполняется, если состояние после завершения выполнения команды гласит "Канал используется". Номер станции находится в диапазоне от 1 до 64.	—	система

- <sup>3</sup> Дополнительная информация о кодах ошибок имеется в руководстве "MELSECNET/10 для сетевых систем QnA".
- <sup>4</sup> Если в Jn указано значение 254, выбран номер сети 254.

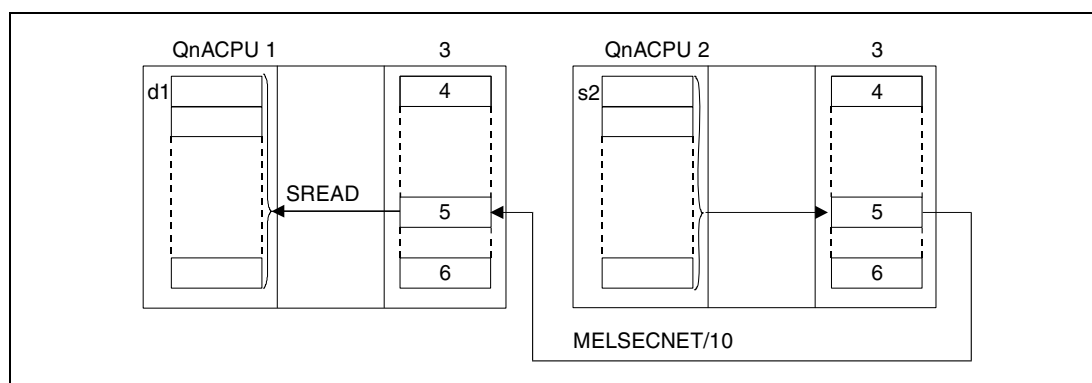
**Принцип действия**

**Считывание данных из словных операндов других станций**

**SREAD Команда считывания**

Команда SREAD считывает данные, хранящиеся начиная с s2 в станции, подключенной к сети MELSECNET/10. Номер станции и номер сети указаны в управляющих данных. Считанные из станции данные сохраняются в хост-станции, начиная с d1.

По окончании операции считывания в хост-станции устанавливается операнд d2, а в целевой станции – операнд d3.



- 1 хост-станция
- 2 целевая станция
- 3 модуль сетевой коммуникации
- 4 канал 1
- 5 канал n
- 6 канал 8

С помощью релейной станции и установленных параметров маршрутизации можно обращаться и к станциям в других сетях.

Команды коммуникации не могут выполняться одновременно из нескольких источников с общим доступом к одним и тем же каналам. При одновременном выполнении команд из двух или более источников выполнение других команд коммуникации предотвращается с помощью режима квитирования (handshake) обеих активных станций.

Состояние выполнения и состояние после завершения (нормальное, не нормальное) команды SREAD можно проверить с помощью:

- флага (●<sup>5</sup>) используемого коммуникационного канала,
- операндов, которые по окончании операции считывания устанавливаются в хост-станции (d2) и целевой станции (d3), и
- индикации состояния после завершения операции (безошибочное или ошибочное завершение передачи) ((d2) + 1)

Проверка выполняется следующим образом:

Флаг коммуникационного канала

Этот флаг устанавливается во время выполнения команды SREAD. Флаг сбрасывается при обработке команды END того цикла, в котором была завершена операция считывания.

Операнд хост-станции, показывающий завершение операции считывания

Этот операнд устанавливается при обработке команды END того цикла, в котором завершена операция считывания. При обработке следующей команды END этот операнд снова сбрасывается.

Индикация состояния после завершения считывания

Этот операнд устанавливается в зависимости от результата после завершения операции считывания.

При завершении нормальной (безошибочной) передачи операнд остается сброшенным.

При завершении ошибочной передачи этот операнд устанавливается при обработке команды END того цикла, в котором была завершена обработка команды SREAD. При обработке следующей команды END этот операнд снова сбрасывается.

Операнд целевой станции, показывающий завершение операции считывания

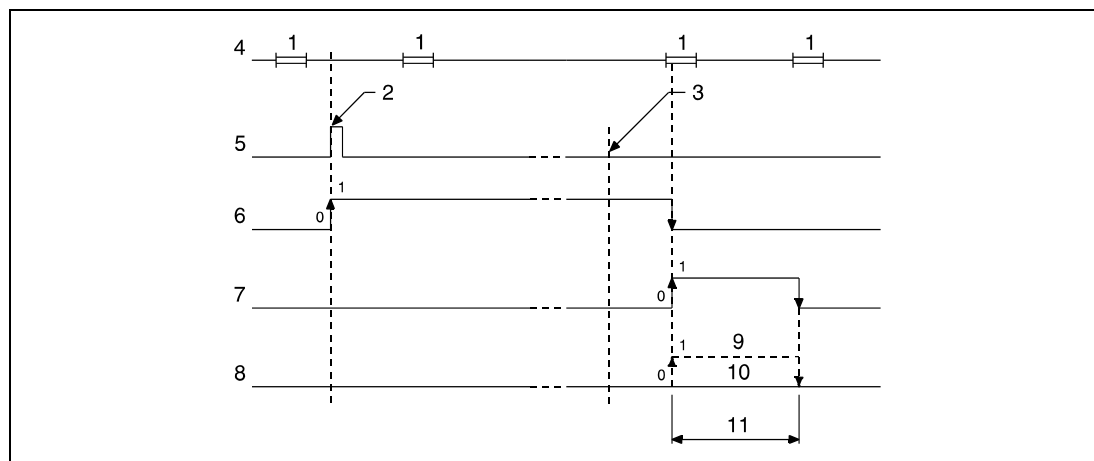
Этот операнд устанавливается при обработке команды END того цикла, в котором была завершена передача данных. При обработке следующей команды END этот операнд снова сбрасывается.

#### ПРИМЕЧАНИЕ

●<sup>5</sup> В следующей таблице пояснена взаимосвязь между номерами каналов и соответствующими флагами коммуникационных каналов.

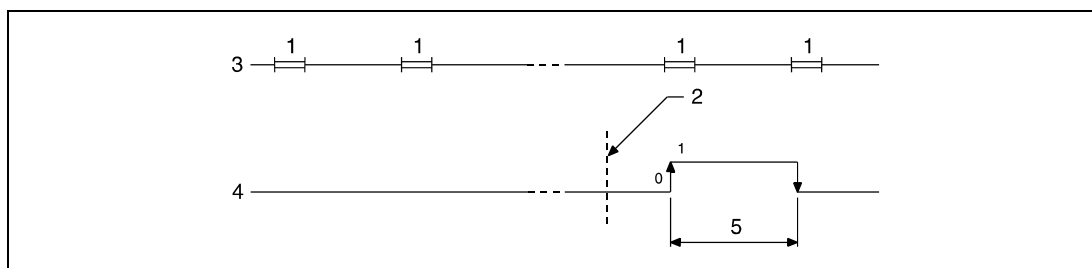
Номер канала	1	2	3	4	5	6	7	8
Флаг коммуникационного канала	SB30	SB32	SB34	SB36	SB38	SB40	SB42	SB44

На рисунке ниже изображены операции хост-станции во время выполнения команды SREAD.



- <sup>1</sup> обработка команды END
- <sup>2</sup> выполнение команды SREAD
- <sup>3</sup> завершение операции
- <sup>4</sup> программа хост-станции
- <sup>5</sup> команда SREAD
- <sup>6</sup> флаг коммуникационного канала
- <sup>7</sup> операнд хост-станции, устанавливаемый по окончании операции ((d2)
- <sup>8</sup> индикация состояния после завершения операции ((d2) + 1)
- <sup>9</sup> завершение ненормальной (ошибочной) передачи
- <sup>10</sup> завершение нормальной (безошибочной) передачи
- <sup>11</sup> один цикл

На рисунке ниже показаны операции целевой станции во время выполнения команды SREAD.



- <sup>1</sup> обработка команды END
- <sup>2</sup> завершение операции
- <sup>3</sup> программа целевой станции
- <sup>4</sup> операнд целевой станции (d3), устанавливаемый по окончании операции
- <sup>5</sup> один цикл

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Содержимое управляющих данных находится вне допустимого диапазона (код ошибки 4100).
- Сеть с номером Jn не соединена со станцией (код ошибки 4102).
- Модуль по адресу ввода-вывода Un не является модулем сетевой коммуникации (код ошибки 2111).

8.6.3 WRITE

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (сист., польз.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	SM0	10	
s2	—	●	●	—	—	—	—	—			
d1	●	●	●	—	—	—	—	—			
d2	●	●	●	—	—	—	—	—			

GX IEC Developer

Язык IL для MELSEC	Рел.-конт. схема (язык LD)	Язык IL по IEC																				
<table border="1" style="width: 100%;"> <tr> <td style="text-align: center; width: 15%;">MELSEC</td> <td style="width: 15%;">JP.WRITE</td> <td style="width: 15%;">Jn</td> <td style="width: 15%;">s1</td> <td style="width: 15%;">s2</td> <td style="width: 15%;">d1</td> <td style="width: 15%;">d2</td> </tr> <tr> <td></td> <td>GP.WRITE</td> <td>Un</td> <td>s1</td> <td>s2</td> <td>d1</td> <td>d2</td> </tr> </table>	MELSEC	JP.WRITE	Jn	s1	s2	d1	d2		GP.WRITE	Un	s1	s2	d1	d2		<table border="1" style="width: 100%;"> <tr> <td style="width: 15%;"></td> <td style="width: 15%;">WRITE_JP_M</td> <td style="width: 15%;">Jn , s1 , s2 , d1 , d2</td> </tr> <tr> <td></td> <td>WRITE_UP_M</td> <td>Un , s1 , s2 , d1 , d2</td> </tr> </table>		WRITE_JP_M	Jn , s1 , s2 , d1 , d2		WRITE_UP_M	Un , s1 , s2 , d1 , d2
MELSEC	JP.WRITE	Jn	s1	s2	d1	d2																
	GP.WRITE	Un	s1	s2	d1	d2																
	WRITE_JP_M	Jn , s1 , s2 , d1 , d2																				
	WRITE_UP_M	Un , s1 , s2 , d1 , d2																				

Переменные

Операнд	Значение	Тип данных	
		MELSEC	IEC
Jn	Номер сети хост-станции ● <sup>1</sup>		
Un	Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции ● <sup>2</sup>	BIN, 16 бит	ANY16
s1	Первый операнд хост-станции, в котором сохранены управляющие данные.	адрес	массив [1..18] данных типа ANY16
s2	Первый операнд станции, в котором сохранены записываемые данные.		ANY16
d1	Первый операнд целевой станции, в который записываются данные.		
d2	Операнд, устанавливаемый после полной обработки команды на один программный цикл.	бит	BOOL



**ПРИМЕЧАНИЯ**

- <sup>1</sup> Номер сети хост-станции должен находиться в диапазоне от 1 до 239.  
Сеть номер 254 с помощью настроек конфигурируется так, чтобы другие станции могли обращаться к активной станции.
- <sup>2</sup> Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции должен находиться в диапазоне от 0 до FFh. Следует иметь в виду, что в Un компилятор ожидает шестнадцатеричного указания адреса. Десятичный адрес автоматически преобразуется в шестнадцатеричное число.

Выполнение команды WRITE возможно только в том случае, если целевая станция построена на процессоре QnA.

При использовании процессора серии "A" в сети MELSECNET/10 команду WRITE применять не следует.

Указывать адрес "FFh" (все станции в целевой сети) в команде WRITE можно только для сетей, к которым подключены только станции на центральных процессорах QnA. Указание адреса "FFh" в смешанных сетях, в которых применяются станции с процессорами серий QnA и "A", не возможно.

## Обзор операндов управляющих данных

Операнд	Значение	Функция	Диапазон	Устанавливает
(s1) + 0 Array_s1[1]	Режим выполнения	Подтверждение завершения передачи = установка бита 0 на 0 Без подтверждения завершения передачи = установка бита 0 на 1	0000h 0001h 0080h 0081h	пользователь
	Режим завершения обработки ошибки	Сохранение момента времени, в который была завершена обработка ошибки: – без сохранения данных времени, бит 7 (b7) = 0 – сохранение данных времени, бит 7 (b7) = 1 (данные времени начиная с (s1) + 10 (Array_s1[11]))		
(s1) + 1 Array_s1[2]	Состояние после завершения обработки команды	Индикация состояния по окончании обработки команды: – 0 = ошибок нет (нормальное завершение обработки) – не равно 0: = код ошибки ● <sup>3</sup>	—	система
(s1) + 2 Array_s1[3]	Канал, используемый хост-станцией	Указание канала, который будет применять хост-станция.	от 1 до 8	пользователь
(s1) + 3 Array_s1[4]	Фиктивный операнд	Не используется.	0	—
(s1) + 4 Array_s1[5]	Номер сети целевой станции	Указание номера сети целевой станции	от 1 до 239 254 ● <sup>4</sup>	пользователь
(s1) + 5 Array_s1[6]	Номер целевой станции	Указание номера целевой станции	Номер станции: от 1 до 64 Групповая адресация: от 81 н до 89н Все станции в целевой сети: FFh	пользователь
(s1) + 6 Array_s1[7]	Фиктивный операнд	Не используется.	—	—
(s1) + 7 Array_s1[8]	Количество попыток передачи	Количество попыток, которое требуется предпринять, чтобы полностью передать данные за контрольное время, заданное в (s1) + 8 (Array_s1[9]). Эта опция действует только в том случае, если режим выполнения ((s1) + 0, Array_s1[1]) установлен (1).	от 1 до 15	пользователь
	Количество выполненных попыток передачи	Сохранение выполненных попыток передачи		
(s1) + 8 Array_s1[9]	Настройка сторожевого таймера для контроля времени приема	Контрольное время для операции в секундах. Если операция не завершается за указанное время, передача повторяется количество раз, указанное в (s1) + 7 (Array_s1[8]).	от 1 до 32767 0 = 10 с (фиксированная настройка) Эта опция действует только в том случае, если режим выполнения установлен (1)	пользователь
(s1) + 9 Array_s1[10]	Длина передаваемых данных	Количество записываемых блоков данных	от 1 до 480	пользователь
(s1) + 10 Array_s1[11]	Фиктивный операнд	Не используется.	—	—
(s1) + 11 Array_s1[12]	Флаг сохранения данных времени (устанавливается только при наличии ошибки)	Этот флаг сохраняет состояние опции сохранения данных времени, указанной в (s1) + 0 (Array_s1[1]); без сохранения данных времени = 0 сохранение данных времени = 1.	—	система

Операнд	Значение	Функция	Диапазон	Устанавливает
(s1) + 12 Array_s1[13]	Данные времени (устанавливаются только при наличии ошибки)	старший байт = год (от 0 до 99) младший байт = месяц (от 1 до 12)	—	система
(s1) + 13 Array_s1[14]		старший байт = день (от 1 до 31) младший байт = час (от 0 до 23)		
(s1) + 14 Array_s1[15]		старший байт = минута (от 0 до 59) младший байт = секунда (от 0 до 59)		
(s1) + 15 Array_s1[16]		старший байт = 00h младший байт = день недели (от 0 до 6) (воскресенье = 0, суббота = 6)		
(s1) + 16 Array_s1[17]	Номер сети, в которой возникла ошибка	Сохранение номера сети станции, в которой возникла ошибка. Номер сети находится в диапазоне от 1 до 239.	—	система
(s1) + 17 Array_s1[18]	Номер станции, в которой возникла ошибка	Сохранение номера станции, в которой возникла ошибка. Сохранение не выполняется, если состояние после завершения выполнения команды гласит "Канал используется". Номер станции находится в диапазоне от 1 до 64.	—	система

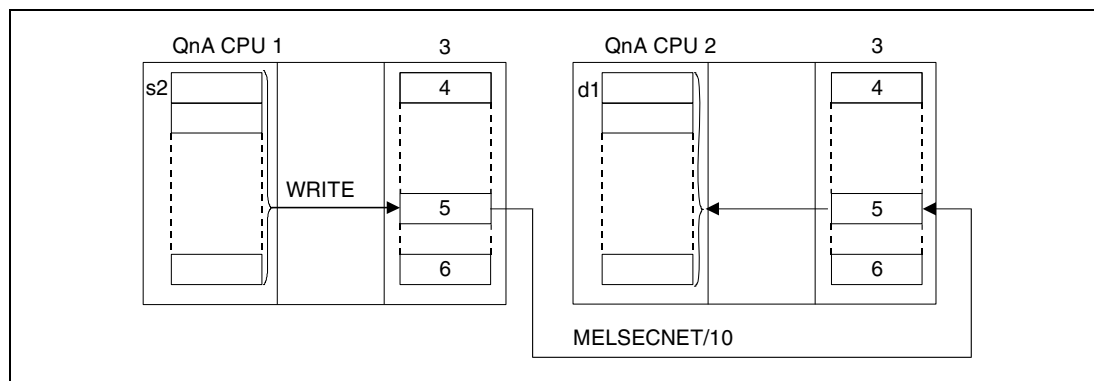
●<sup>3</sup> Дополнительная информация о кодах ошибок имеется в руководстве "MELSECNET/10 для сетевых систем QnA".

●<sup>4</sup> Если в Jn указано значение 254, выбран номер сети 254.

**Принцип действия****Запись данных в словные операнды других станций****WRITE Команда записи**

Команда WRITE записывает данные, хранящиеся начиная с s2 в хост-станции, в станцию, подключенную к сети MELSECNET/10. Номер станции и номер сети указаны в управляющих данных. Данные сохраняются в целевой станции, начиная с d1.

По окончании операции записи устанавливается операнд d2.



- 1 хост-станция
- 2 целевая станция
- 3 модуль сетевой коммуникации
- 4 канал 1
- 5 канал n
- 6 канал 8

С помощью релейной станции и установленных параметров маршрутизации можно обращаться и к станциям в других сетях.

Команды коммуникации не могут выполняться одновременно из нескольких источников с общим доступом к одним и тем же каналам. При одновременном выполнении команд из двух или более источников выполнение других команд коммуникации предотвращается благодаря квитированию связи (handshake) между двумя активными станциями.

Состояние выполнения и состояние после завершения (нормальное, ненормальное) команды WRITE можно проверить с помощью:

- флага (●<sup>5</sup>) используемого коммуникационного канала,
- операнда, устанавливаемого по окончании операции записи (d2), и
- индикации состояния после завершения операции (безошибочное или ошибочное завершение передачи) ((d2) + 1)

Проверка выполняется следующим образом:

Флаг коммуникационного канала

Этот флаг устанавливается во время выполнения команды WRITE. Флаг сбрасывается при обработке команды END того цикла, в котором была завершена операция записи.

Операнд хост-станции, показывающий завершение операции записи

Этот операнд устанавливается при обработке команды END того цикла, в котором была завершена операция записи. При обработке следующей команды END этот операнд снова сбрасывается.

Индикация состояния после завершения записи

Этот операнд устанавливается в зависимости от результата после завершения операции записи.

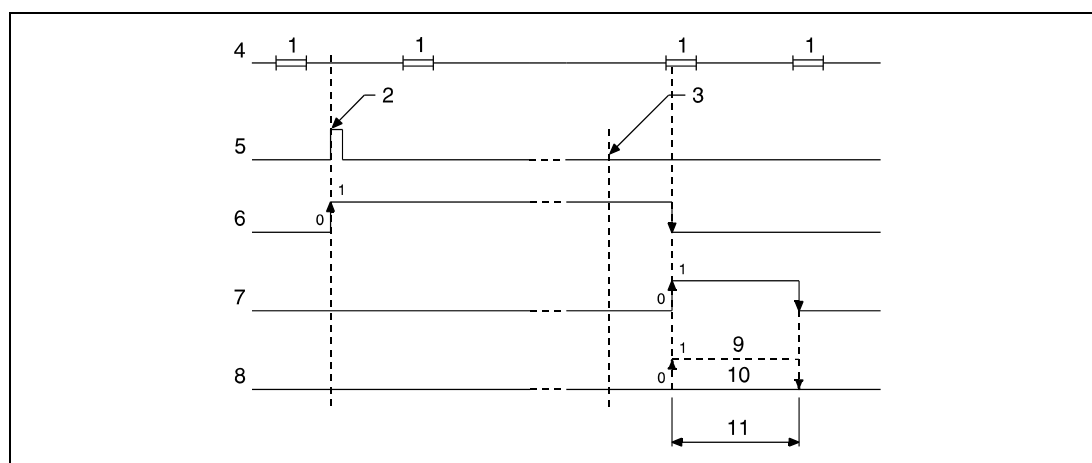
При завершении нормальной (безошибочной) передачи операнд остается сброшенным.

После завершения ненормальной передачи (т. е. передачи, содержащей ошибки) этот операнд устанавливается при обработке команды END того цикла, в котором была завершена обработка команды WRITE. При обработке следующей команды END этот операнд снова сбрасывается.

**ПРИМЕЧАНИЕ** ●<sup>5</sup> В следующей таблице пояснена взаимосвязь между номерами каналов и соответствующими флагами коммуникационных каналов.

Номер канала	1	2	3	4	5	6	7	8
Флаг коммуникационного канала	SB30	SB32	SB34	SB36	SB38	SB3A	SB3B	SB3D

На рисунке ниже изображены операции хост-станции во время выполнения команды WRITE.



- 1 обработка команды END
- 2 выполнение команды WRITE
- 3 завершение операции
- 4 программа хост-станции
- 5 команда WRITE
- 6 флаг коммуникационного канала
- 7 операнд хост-станции, устанавливаемый по окончании операции (d2)
- 8 индикация состояния после завершения операции ((d2) + 1)
- 9 завершение ненормальной (ошибочной) передачи
- 10 завершение нормальной (безошибочной) передачи
- 11 один цикл

**Источники  
ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Содержимое управляющих данных находится вне допустимого диапазона (код ошибки 4100).
- Сеть с номером Jn не соединена со станцией (код ошибки 4102).
- Модуль по адресу ввода-вывода Un не является модулем сетевой коммуникации (код ошибки 2111).

### 8.6.4 SWRITE

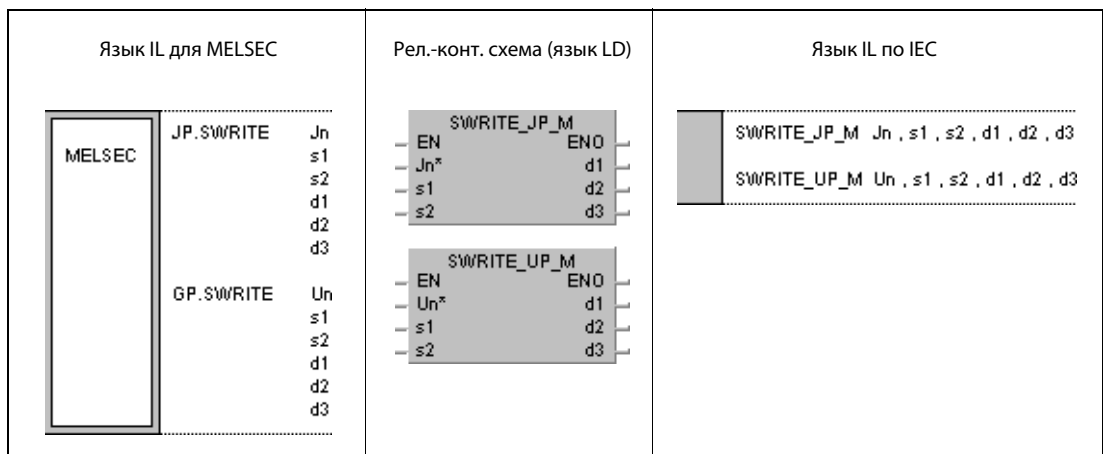
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (сист., польз.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	SM0	11	
s2	—	●	●	—	—	—	—	—			
d1	●	●	●	—	—	—	—	—			
d2	●	●	●	—	—	—	—	—			
d3	●	●	●	—	—	—	—	—			

GX IEC Developer



Переменные

Операнд	Значение	Тип данных	
		MELSEC	IEC
Jn	Номер сети хост-станции ● <sup>1</sup>		
Un	Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции ● <sup>2</sup>	BIN, 16 бит	ANY16
s1	Первый операнд хост-станции, в котором сохранены управляющие данные.	адрес	массив [1..18] данных типа ANY16
s2	Первый операнд станции, в котором сохранены записываемые данные.		ANY16
d1	Первый операнд целевой станции, в который записываются данные.		
d2	Операнд хост-станции, устанавливаемый после полной обработки команды на один программный цикл.	бит	BOOL
d3	Операнд целевой станции, устанавливаемый после полной обработки команды на один программный цикл.		

**ПРИМЕЧАНИЯ**

- <sup>1</sup> Номер сети хост-станции должен находиться в диапазоне от 1 до 239.  
Сеть номер 254 с помощью настроек конфигурируется так, чтобы другие станции могли обращаться к активной станции.
- <sup>2</sup> Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции должен находиться в диапазоне от 0 до FFh. Следует иметь в виду, что в Un компилятор ожидает шестнадцатеричного указания адреса. Десятичный адрес автоматически преобразуется в шестнадцатеричное число.

Выполнение команды SWRITE возможно только в том случае, если целевая станция построена на процессоре QnA.

При использовании процессора серии "A" в сети MELSECNET/10 команду SWRITE применять не следует.

Указывать адрес "FFh" (все станции в целевой сети) в команде SWRITE можно только для сети, к которой подключены только станции на центральных процессорах серии QnA. Указание адреса "FFh" в смешанных сетях, в которых применяются станции с процессорами серий QnA и "A", не возможно.



**Обзор операндов управляющих данных**

Операнд	Значение	Функция	Диапазон	Устанавливает
(s1) + 0 Array_s1[1]	Режим выполнения	Подтверждение завершения передачи = установка бита 0 на 0 Без подтверждения завершения передачи = установка бита 0 на 1	0000h 0001h 0080h 0081h	пользователь
	Режим завершения обработки ошибки	Сохранение момента времени, в который была завершена обработка ошибки: – без сохранения данных времени, бит 7 (b7) = 0 – сохранение данных времени, бит 7 (b7) = 1 (данные времени начиная с (s1) + 10 (Array_s1[11]))		
(s1) + 1 Array_s1[2]	Состояние после завершения обработки команды	Индикация состояния по окончании обработки команды: – 0 = ошибок нет (нормальное завершение обработки) – не равно 0: = код ошибки ● <sup>3</sup>	—	система
(s1) + 2 Array_s1[3]	Канал, используемый хост-станцией	Указание канала, который будет применять хост-станция.	от 1 до 8	пользователь
(s1) + 3 Array_s1[4]	Фиктивный операнд	Не используется.	0	—
(s1) + 4 Array_s1[5]	Номер сети целевой станции	Указание номера сети целевой станции	от 1 до 239 254 ● <sup>4</sup>	пользователь
(s1) + 5 Array_s1[6]	Номер целевой станции	Указание номера целевой станции	Номер станции: от 1 до 64 Групповая адресация: от 81 н до 89н Все станции в целевой сети: FFh	пользователь
(s1) + 6 Array_s1[7]	Фиктивный операнд	Не используется.	—	—
(s1) + 7 Array_s1[8]	Количество попыток передачи	Количество попыток, которое требуется предпринять, чтобы полностью передать данные за контрольное время, заданное в (s1) + 8 (Array_s1[9]). Эта опция действует только в том случае, если режим выполнения ((s1)+0, Array_s1[1]) установлен (1).	от 1 до 15	пользователь
	Количество выполненных попыток передачи	Сохранение выполненных попыток передачи		
(s1) + 8 Array_s1[9]	Настройка сторожевого таймера для контроля времени приема	Контрольное время для операции в секундах. Если операция не завершается за указанное время, передача повторяется количество раз, указанное в (s1) + 7 (Array_s1[8]).	от 1 до 32767 0 = 10 с (фиксированная настройка) Эта опция действует только в том случае, если режим выполнения установлен (1)	пользователь
(s1) + 9 Array_s1[10]	Длина принимаемых данных	Количество записываемых блоков данных.	от 1 до 480	пользователь
(s1) + 10 Array_s1[11]	Фиктивный операнд	Не используется.	—	—
(s1) + 11 Array_s1[12]	Флаг сохранения данных времени (устанавливается только при наличии ошибки)	Этот флаг сохраняет состояние опции сохранения данных времени, указанной в (s1) + 0 (Array_s1[1]); без сохранения данных времени = 0 сохранение данных времени = 1	—	система

Операнд	Значение	Функция	Диапазон	Устанавливает
(s1) + 12 Array_s1[13]	Данные времени (устанавливаются только при наличии ошибки)	старший байт = год (от 0 до 99) младший байт = месяц (от 1 до 12)	—	система
(s1) + 13 Array_s1[14]		старший байт = день (от 1 до 31) младший байт = час (от 0 до 23)		
(s1) + 14 Array_s1[15]		старший байт = минута (от 0 до 59) младший байт = секунда (от 0 до 59)		
(s1) + 15 Array_s1[16]		старший байт = 00н младший байт = день недели (от 0 до 6) (воскресенье = 0, суббота = 6)		
(s1) + 16 Array_s1[17]	Номер сети, в которой возникла ошибка	Сохранение номера сети станции, в которой возникла ошибка. Номер сети находится в диапазоне от 1 до 239.	—	система
(s1) + 17 Array_s1[18]	Номер станции, в которой возникла ошибка	Сохранение номера станции, в которой возникла ошибка. Сохранение не выполняется, если состояние после завершения выполнения команды гласит "Канал используется". Номер станции находится в диапазоне от 1 до 64.	—	система

●<sup>3</sup> Дополнительная информация о кодах ошибок имеется в руководстве "MELSECNET/10 для сетевых систем QnA".

●<sup>4</sup> Если в Jn указано значение 254, выбран номер сети 254.

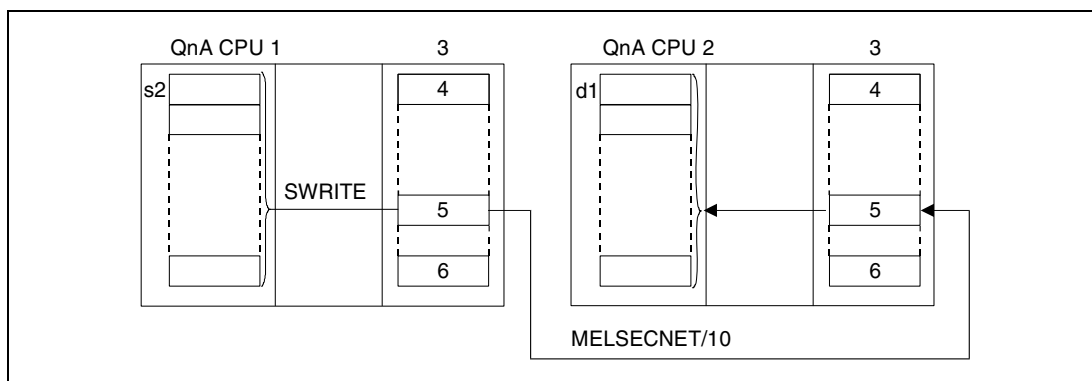
**Принцип действия**

**Запись данных в словные операнды других станций**

**SWRITE Команда записи**

Команда SWRITE записывает данные, хранящиеся начиная с s2 в хост-станции, в станцию, подключенную к сети MELSECNET/10. Номер станции и номер сети указаны в управляющих данных. Данные сохраняются в целевой станции, начиная с d1.

По окончании операции записи в хост-станции устанавливается операнд d2, а в целевой станции – операнд d3.



- 1 хост-станция
- 2 целевая станция
- 3 модуль сетевой коммуникации
- 4 канал 1
- 5 канал n
- 6 канал 8

С помощью релейной станции и установленных параметров маршрутизации можно обращаться и к станциям в других сетях.

Команды коммуникации не могут выполняться одновременно из нескольких источников с общим доступом к одним и тем же каналам. При одновременном выполнении команд из двух или более источников выполнение других команд коммуникации предотвращается с помощью режима квитирования (handshake) обеих активных станций.

Состояние выполнения и состояние после завершения (нормальное, ненормальное) команды SWRITE можно проверять с помощью:

- флага (●<sup>5</sup>) используемого коммуникационного канала,
- операнда, устанавливаемого по окончании операции записи в хост-станции (d2) и целевой станции (d3)
- индикации состояния после завершения операции (безошибочная или ошибочная передача) ((d2) + 1)

Проверка выполняется следующим образом:

Флаг коммуникационного канала

Этот флаг устанавливается во время выполнения команды SWRITE. Флаг сбрасывается при обработке команды END того цикла, в котором была завершена операция записи.

Операнд хост-станции, показывающий завершение операции записи

Этот операнд устанавливается при обработке команды END того цикла, в котором была завершена операция записи. При обработке следующей команды END этот операнд снова сбрасывается.

Индикация состояния после завершения записи

Этот операнд устанавливается в зависимости от результата после завершения операции записи.

При завершении нормальной (безошибочной) передачи операнд остается сброшенным.

При завершении ошибочной передачи этот операнд устанавливается при обработке команды END того цикла, в котором была завершена обработка команды SWRITE. При обработке следующей команды END этот операнд снова сбрасывается.

Операнд целевой станции, показывающий завершение операции записи

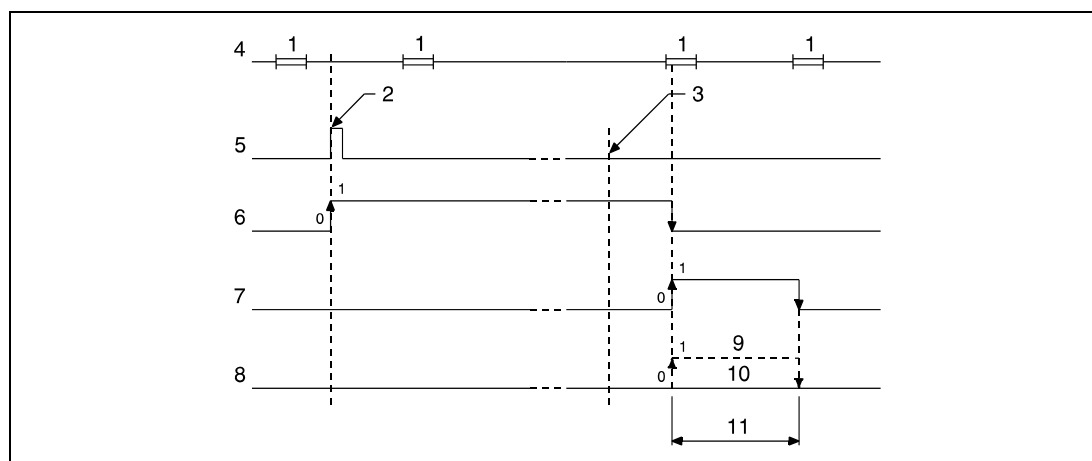
Этот операнд устанавливается при обработке команды END того цикла, в котором была завершена передача данных, указанных в команде SWRITE. При обработке следующей команды END этот операнд снова сбрасывается.

**ПРИМЕЧАНИЕ**

●<sup>5</sup> В следующей таблице пояснена взаимосвязь между номерами каналов и соответствующими флагами коммуникационных каналов.

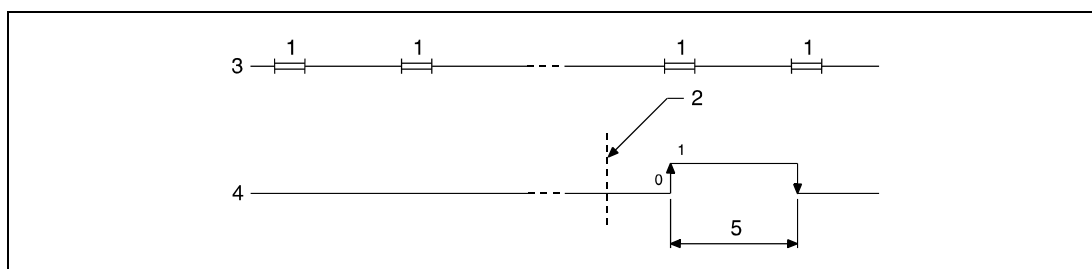
Номер канала	1	2	3	4	5	6	7	8
Флаг коммуникационного канала	SB30	SB32	SB34	SB36	SB38	SB40	SB42	SB44

На рисунке ниже изображены операции хост-станции во время выполнения команды SWRITE.



- 1 обработка команды END
- 2 выполнение команды SWRITE
- 3 завершение операции
- 4 программа хост-станции
- 5 команда SWRITE
- 6 флаг коммуникационного канала
- 7 операнд хост-станции, устанавливаемый по окончании операции ((d2)
- 8 индикация состояния после завершения операции ((d2) + 1)
- 9 завершение ненормальной (ошибочной) передачи
- 10 завершение нормальной (безошибочной) передачи
- 11 один цикл

На рисунке ниже показаны операции целевой станции во время выполнения команды SWRITE.



- 1 обработка команды END
- 2 завершение операции
- 3 программа целевой станции
- 4 операнд целевой станции (d3), устанавливаемый по окончании операции
- 5 один цикл

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Содержимое управляющих данных находится вне допустимого диапазона (код ошибки 4100).
- Сеть с номером Jn не соединена со станцией (код ошибки 4102).
- Модуль по адресу ввода-вывода Un не является модулем сетевой коммуникации (код ошибки 2111).

## 8.6.5 SEND

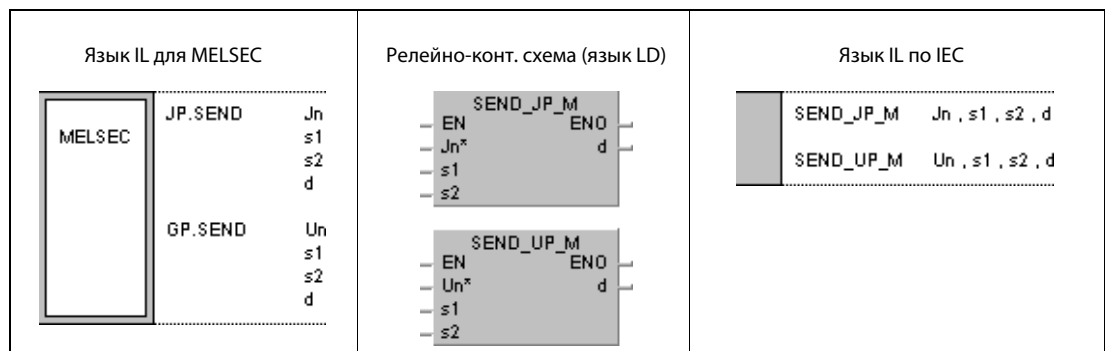
## Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

## Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (сист., польз.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	SMO	8	
s2	—	●	●	—	—	—	—	—			
d	●	●	●	—	—	—	—	—			

## GX IEC Developer



## Переменные

Операнд	Значение	Тип данных	
		MELSEC	IEC
Jn	Номер сети хост-станции ● <sup>1</sup>	BIN, 16 бит	ANY16
Un	Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции ● <sup>2</sup>		
s1	Первый адрес операнда, в котором сохранены управляющие данные.	адрес	массив [1..18] данных типа ANY16
s2	Первый адрес операнда, в котором сохранены передаваемые данные.		ANY16
d	Операнд, устанавливаемый после полной обработки команды на один программный цикл.	бит	BOOL

## ПРИМЕЧАНИЯ

- <sup>1</sup> Номер сети хост-станции должен находиться в диапазоне от 1 до 239. Сеть номер 254 с помощью настроек конфигурируется так, чтобы другие станции могли обращаться к активной станции.
- <sup>2</sup> Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции должен находиться в диапазоне от 0 до FEn. Следует иметь в виду, что в Un компилятор ожидает шестнадцатеричного указания адреса. Десятичный адрес автоматически преобразуется в шестнадцатеричное число.

**Обзор операндов управляющих данных**

Операнд	Значение	Функция	Диапазон	Устанавливает
(s1) + 0 Array_s1[1]	Режим выполнения	Подтверждение завершения передачи = установка бита 0 на 0 Без подтверждения завершения передачи = установка бита 0 на 1	0000h 0001h 0080h 0081h	пользователь
	Режим завершения обработки ошибки	Сохранение момента времени, в который была завершена обработка ошибки: – без сохранения данных времени, бит 7 (b7) = 0 – сохранение данных времени, бит 7 (b7) = 1 (данные времени начиная с (s1) + 11 (Array_s1[12]))		
(s1) + 1 Array_s1[2]	Состояние после завершения обработки команды	Индикация состояния по окончании обработки команды: – 0 = ошибок нет (нормальное завершение обработки) – не равно 0: = код ошибки ● <sup>3</sup>	—	система
(s1) + 2 Array_s1[3]	Канал, используемый хост-станцией	Указание канала, который будет применять хост-станция.	от 1 до 8	пользователь
(s1) + 3 Array_s1[4]	Канал, используемый целевой станцией	Указание номера сети целевой станции	от 1 до 8	пользователь
(s1) + 4 Array_s1[5]	Номер сети целевой станции	Указание номера сети целевой станции	от 1 до 239 254 ● <sup>4</sup>	пользователь
(s1) + 5 Array_s1[6]	Номер целевой станции	Указание номера целевой станции. (адресация в диапазоне от "81h" до "89h" возможна только при сброшенном режиме выполнения).	Номер станции: от 1 до 64 Групповая адресация: от 81h до 89h Все станции в целевой сети: FFh	пользователь
(s1) + 6 Array_s1[7]	Фиктивный операнд	Не используется.	—	—
(s1) + 7 Array_s1[8]	Количество попыток передачи	Количество попыток, которое требуется предпринять, чтобы полностью передать данные за контрольное время, заданное в (s1) + 8 (Array_s1[9]). Эта опция действует только в том случае, если режим выполнения ((s1)+0, Array_s1[1]) установлен (1).	от 1 до 15	пользователь
	Количество выполненных попыток передачи	Сохранение выполненных попыток передачи.		
(s1) + 8 Array_s1[9]	Настройка сторожевого таймера для контроля времени приема	Контрольное время для операции в секундах. Если операция не завершается за указанное время, передача повторяется количество раз, указанное в (s1) + 7 (Array_s1[8]).	от 1 до 32767 0 = 10 с (фиксированная настройка) Эта опция действует только в том случае, если режим выполнения установлен (1)	пользователь
(s1) + 9 Array_s1[10]	Длина передаваемых данных	Указание количества передаваемых блоков данных.	от 1 до 480	пользователь
(s1) + 10 Array_s1[11]	Фиктивный операнд	Не используется.	—	—
(s1) + 11 Array_s1[12]	Флаг сохранения данных времени (устанавливается только при наличии ошибки)	Этот флаг сохраняет состояние опции сохранения данных времени, указанной в (s1) + 0 (Array_s1[1]); без сохранения данных времени = 0 сохранение данных времени = 1	—	система

Операнд	Значение	Функция	Диапазон	Устанавливает
(s1) + 12 Array_s1[13]	Данные времени (устанавливаются только при наличии ошибки)	старший байт = год (от 0 до 99) младший байт = месяц (от 1 до 12)	—	система
(s1) + 13 Array_s1[14]		старший байт = день (от 1 до 31) младший байт = час (от 0 до 23)		
(s1) + 14 Array_s1[15]		старший байт = минута (от 0 до 59) младший байт = секунда (от 0 до 59)		
(s1) + 15 Array_s1[16]		старший байт = 00н младший байт = день недели (от 0 до 6) (воскресенье = 0, суббота = 6)		
(s1) + 16 Array_s1[17]	Номер сети, в которой возникла ошибка	Сохранение номера сети станции, в которой возникла ошибка. Номер сети находится в диапазоне от 1 до 239.	—	система
(s1) + 17 Array_s1[18]	Номер станции, в которой возникла ошибка	Сохранение номера станции, в которой возникла ошибка. Сохранение не выполняется, если состояние после завершения выполнения команды гласит "Канал используется". Номер станции находится в диапазоне от 1 до 64.	—	система

●<sup>3</sup> Дополнительная информация о кодах ошибок имеется в руководстве "MELSECNET/10 для сетевых систем QnA".

●<sup>4</sup> Если в Jn указано значение 254, выбран номер сети 254.



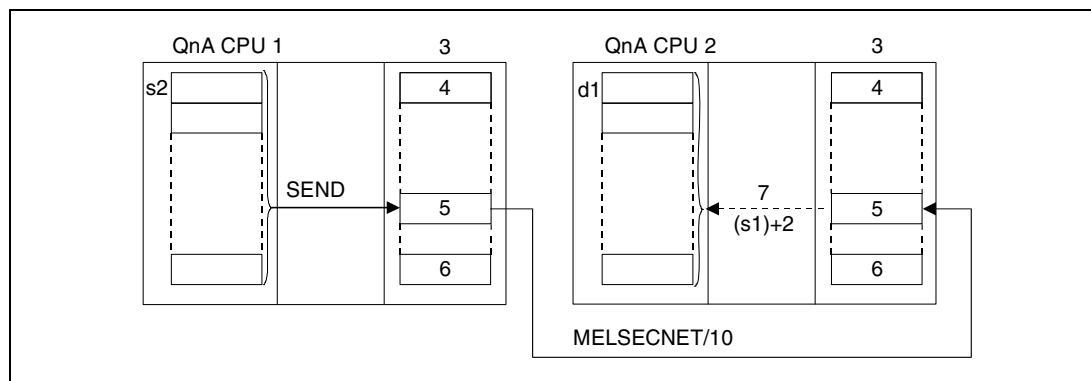
**Принцип действия**

**Передача данных на другие станции**

**SEND Команда передачи**

Команда SEND передает данные, хранящиеся начиная с s2 в хост-станции, в станцию, подключенную к сети MELSECNET/10. Канал передачи указан в (s1) + 2. Номер станции и номер сети указаны в управляющих данных.

По окончании операции записи в целевой станции устанавливается операнд, указанный в d.



- 1 хост-станция
- 2 целевая станция
- 3 модуль сетевой коммуникации
- 4 канал 1
- 5 канал n
- 6 канал 8
- 7 операция считывания выполняется с помощью команды RECV

С помощью релейной станции и при установленных параметрах маршрутизации возможно обращение и к станциям в других сетях.

Команды коммуникации не могут выполняться одновременно из нескольких источников с общим доступом к одним и тем же каналам. При одновременном выполнении команд из двух или более источников выполнение других команд коммуникации предотвращается с помощью режима квитирования (handshake) обеих активных станций.

Состояние выполнения и состояние после завершения (нормальное, ненормальное) команды SEND можно контролировать с помощью:

- флага (●<sup>5</sup>) используемого коммуникационного канала,
- операнда, устанавливаемого по окончании операции (d), и
- индикации состояния после завершения операции (безошибочная или ошибочная передача) (d + 1)

Контроль выполняется следующим образом:

Флаг коммуникационного канала

Этот флаг устанавливается во время выполнения команды SEND. Флаг сбрасывается при обработке команды END того цикла, в котором операция была завершена.

Операнд хост-станции, показывающий завершение операции

Этот операнд устанавливается при обработке команды END того цикла, в котором была завершена операция. При обработке следующей команды END этот операнд снова сбрасывается.

Индикация состояния после завершения операции

Этот операнд устанавливается в зависимости от результата после завершения операции.

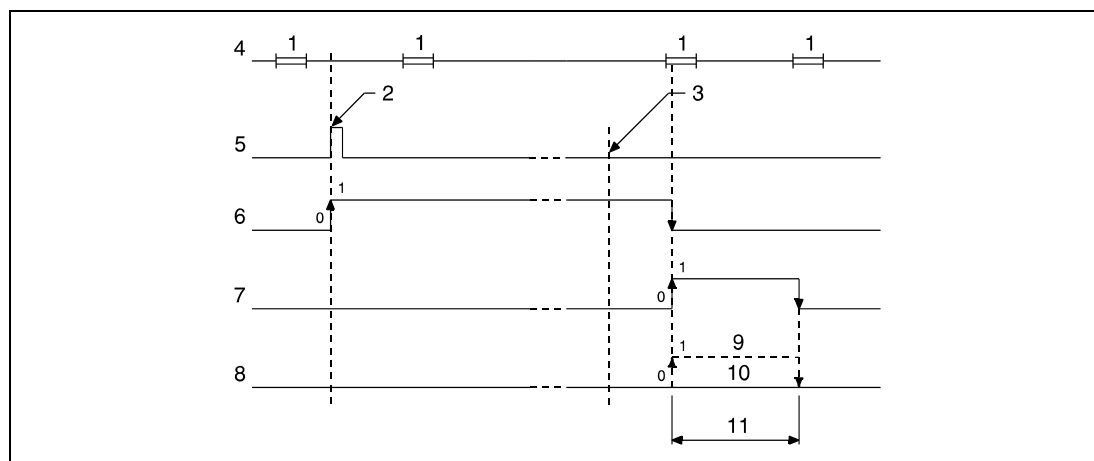
При завершении нормальной (безошибочной) передачи операнд остается сброшенным.

После завершения ненормальной передачи (т. е. передачи, содержащей ошибки) этот операнд устанавливается при обработке команды END того цикла, в котором была завершена обработка команды SEND. При обработке следующей команды END этот операнд снова сбрасывается.

**ПРИМЕЧАНИЕ** ●<sup>5</sup> В следующей таблице пояснена взаимосвязь между номерами каналов и соответствующими флагами коммуникационных каналов.

Номер канала	1	2	3	4	5	6	7	8
Флаг коммуникационного канала	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

На рисунке ниже изображены операции хост-станции во время выполнения команды SEND.



- 1 обработка команды END
- 2 выполнение команды SEND
- 3 завершение операции
- 4 программа хост-станции
- 5 команда SEND
- 6 флаг коммуникационного канала
- 7 операнд хост-станции, устанавливаемый по окончании операции (d)
- 8 индикация состояния после завершения операции (d + 1)
- 9 завершение ненормальной (ошибочной) передачи
- 10 завершение нормальной (безошибочной) передачи
- 11 один цикл

**Источники  
ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

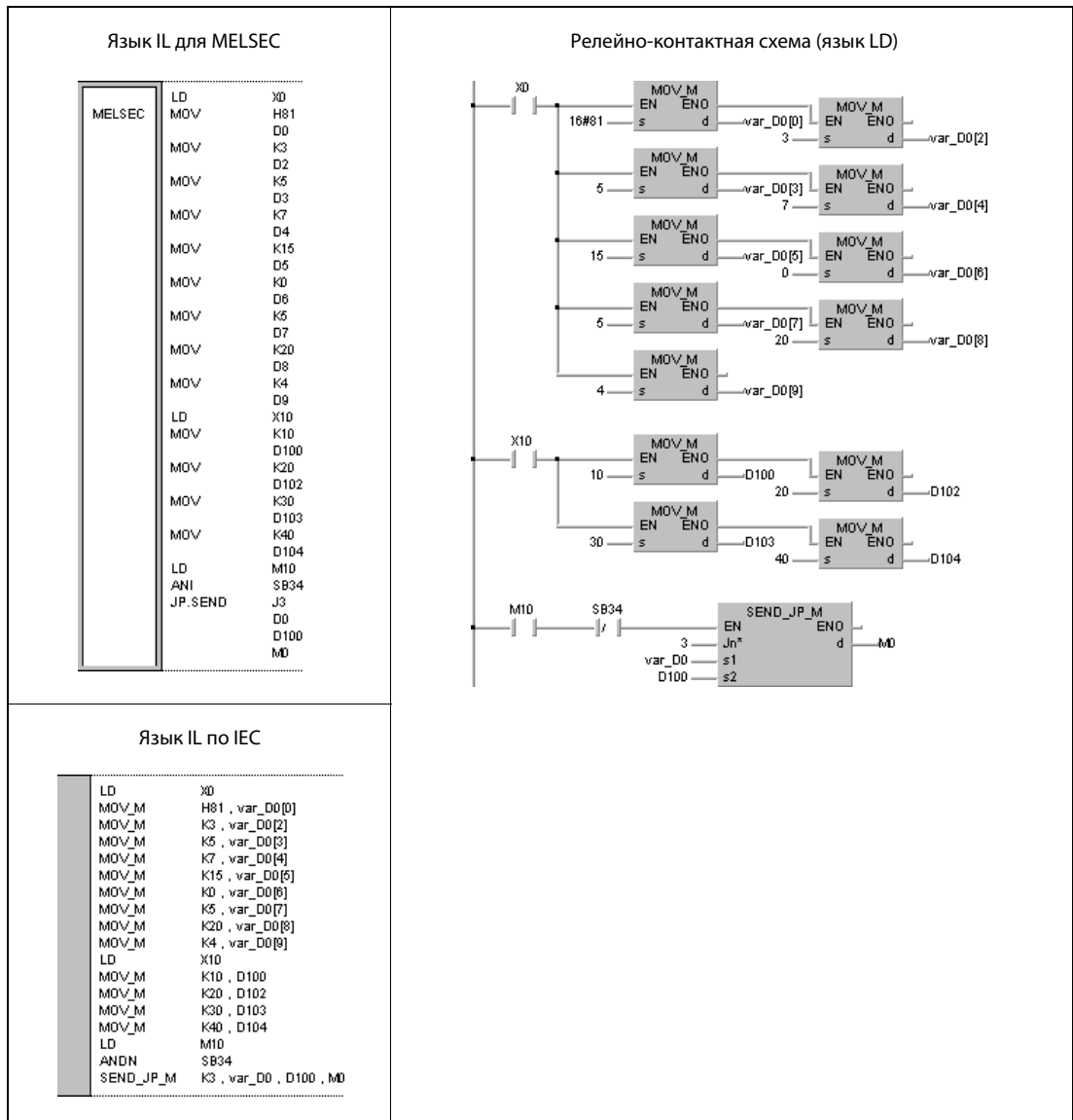
- Содержимое управляющих данных находится вне допустимого диапазона (код ошибки 4100).
- Сеть с номером Jn не соединена со станцией (код ошибки 4102).
- Модуль по адресу ввода-вывода Un не является модулем сетевой коммуникации (код ошибки 2111).

**Пример**

JP.SEND

Следующая программа при положительном фронте M10 передает данные хост-станции в целевую станцию. Выполнение команды SEND заблокировано размыкающим контактом флага SB34. Более подробная информация о хост-станции, целевой станции и применении команд MOV содержится в следующей таблице.

Операнд/команда	Обозначение/функция
Хост-станция	—
Главная сеть (хост)	7
Главный канал (хост)	3
Флаг коммуникационного канала	SB34
Целевая станция	15
Целевая сеть	5
Целевой канал	5
1-я команда MOV	Установка входного условия и данных времени
2-я команда MOV	Установка канала хост-станции
3-я команда MOV	Установка канала целевой станции
4-я команда MOV	Установка номера сети целевой станции
5-я команда MOV	Установка номера целевой станции
6-я команда MOV	—
7-я команда MOV	Установка количества попыток передачи
8-я команда MOV	Установка времени сторожевого таймера (20 с)
9-я команда MOV	Установка количества передаваемых блоков (4)
10-я команда MOV	Установка передаваемых данных
11-я команда MOV	
12-я команда MOV	
13-я команда MOV	



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

8.6.6 RECV


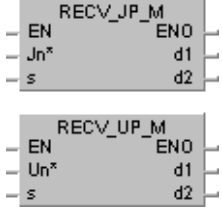
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (сист., польз.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	—	SM0	8
d1	—	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		

GX IEC Developer

Язык IL для MELSEC	Релейно-контактная схема (язык LD)	Язык IL по IEC
 <pre> Jn.RECV  Jn s d1 d2  Un s d1 d2                     </pre>		<pre> RECJV_JP_M  Jn , s , d1 , d2 RECJV_UP_M  Un , s , d1 , d2                     </pre>

Переменные

Операнд	Значение	Тип данных	
		MELSEC	IEC
Jn	Номер сети хост-станции ● <sup>1</sup>	BIN, 16 бит	ANY16
Un	Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции ● <sup>2</sup>		
s	Первый адрес операнда, в котором сохранены управляющие данные.	адрес	массив [1..16] данных типа ANY16
d1	Первый адрес операнда, в котором сохраняются переданные данные.		ANY16
d2	Операнд, устанавливаемый после полной обработки команды на один программный цикл.	бит	BOOL

ПРИМЕЧАНИЯ

- <sup>1</sup> Номер сети хост-станции должен находиться в диапазоне от 1 до 239. Сеть номер 254 с помощью настроек конфигурируется так, чтобы другие станции могли обращаться к активной станции.
- <sup>2</sup> Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции должен находиться в диапазоне от 0 до FEn. Следует иметь в виду, что в Un компилятор ожидает шестнадцатеричного указания адреса. Десятичный адрес автоматически преобразуется в шестнадцатеричное число.

**Обзор операндов управляющих данных**

Операнд	Значение	Функция	Диапазон	Устанавливает
(s1) + 0 Array_s1[1]	Режим выполнения	Ожидание данных (в этом режиме происходит периодическое ожидание данных с фиксированной настройкой времени): (бит 0 (b0) = 0, фиксированная настройка)	0000h 0080h	пользователь
	Режим завершения обработки ошибки	Сохранение момента времени, в который была завершена обработка ошибки: – без сохранения данных времени, бит 7 (b7) = 0 – сохранение данных времени, бит 7 (b7) = 1 (данные времени начиная с (s1) + 11 (Array_s1[12]))		
(s1) + 1 Array_s1[2]	Состояние после завершения обработки команды	Индикация состояния по окончании обработки команды: – 0 = ошибок нет (нормальное завершение обработки) – не равно 0: = код ошибки ● <sup>3</sup>	—	система
(s1) + 2 Array_s1[3]	Канал, используемый хост-станцией	Указание канала, в котором сохранены принимаемые данные.	от 1 до 8	пользователь
(s1) + 3 Array_s1[4]	Каналы, используемые целевыми станциями	Сохранение канала, используемого передающей станцией.	от 1 до 8	система
(s1) + 4 Array_s1[5]	Номер сети целевой станции	Сохранение номера сети передающей станции	от 1 до 239	система
(s1) + 5 Array_s1[6]	Номер целевой станции	Сохранение номера передающей станции	от 1 до 64	система
(s1) + 6 Array_s1[7]	Фиктивный операнд	Не используется.	—	—
(s1) + 7 Array_s1[8]	Фиктивный операнд	Не используется.	—	—
(s1) + 8 Array_s1[9]	Настройка сторожевого таймера для контроля времени приема	Контрольное время для операции в секундах	от 1 до 32767 0 = 10 с (фиксированная настройка) Эта опция действует только в том случае, если режим выполнения установлен (1)	пользователь
(s1) + 9 Array_s1[10]	Длина принимаемых данных	Сохранение количества принятых блоков данных.	от 1 до 480	система
(s1) + 10 Array_s1[11]	Фиктивный операнд	Не используется.	—	—
(s1) + 11 Array_s1[12]	Флаг сохранения данных времени (устанавливается только при наличии ошибки)	Этот флаг сохраняет состояние опции сохранения данных времени, указанной в (s1) + 0 (Array_s1[1]): без сохранения данных времени = 0 сохранение данных времени = 1.	—	система
(s1) + 12 Array_s1[13]	Данные времени (устанавливаются только при наличии ошибки)	старший байт = год (от 0 до 99) младший байт = месяц (от 1 до 12)	—	система
(s1) + 13 Array_s1[14]		старший байт = день (от 1 до 31) младший байт = час (от 0 до 23)		
(s1) + 14 Array_s1[15]		старший байт = минута (от 0 до 59) младший байт = секунда (от 0 до 59)		
(s1) + 15 Array_s1[16]		старший байт = 00h младший байт = день недели (от 0 до 6) (воскресенье = 0, суббота = 6)		

●<sup>3</sup> Дополнительная информация о кодах ошибок имеется в руководстве "MELSECNET/10 для сетевых систем QnA".

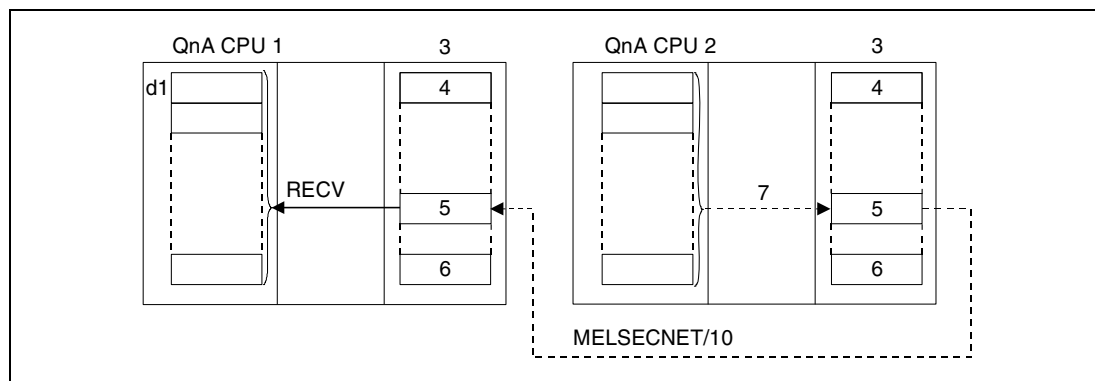
**Принцип действия**

**Прием данных от других станций**

**RECV Команда приема**

Команда RECV принимает данные, переданные с помощью команды SEND станцией, подключенной к сети MELSECNET/10. Номер станции и номер сети указаны в управляющих данных. Данные сохраняются, начиная с d1.

По окончании операции устанавливается операнд, указанный в d2.



- 1 хост-станция
- 2 целевая станция
- 3 модуль сетевой коммуникации
- 4 канал 1
- 5 канал n
- 6 канал 8
- 7 Операция записи выполняется с помощью команды SEND.

С помощью релейной станции и установленных параметров маршрутизации можно обращаться и к станциям в других сетях.

Команды коммуникации не могут выполняться одновременно из нескольких источников с общим доступом к одним и тем же каналам. При одновременном выполнении команд из двух или более источников выполнение других команд коммуникации предотвращается с помощью режима квитирования (handshake) обеих активных станций.

Состояние выполнения и состояние после завершения (нормальное, ненормальное) команды RECV можно контролировать с помощью:

- флага (●<sup>4</sup>) используемого коммуникационного канала,
- операнда, устанавливаемого по окончании операции (d2), и
- индикации состояния после завершения операции (безошибочная или ошибочная передача) ((d2) + 1)

Контроль выполняется следующим образом:

**Флаг коммуникационного канала**

Этот флаг устанавливается во время выполнения команды RECV. Флаг сбрасывается при обработке команды END того цикла, в котором была завершена операция считывания.

**Операнд хост-станции, показывающий завершение операции**

Этот операнд устанавливается при обработке команды END того цикла, в котором была завершена операция. При обработке следующей команды END этот операнд снова сбрасывается.



Индикация состояния после завершения операции

Этот операнд устанавливается в зависимости от результата после завершения операции.

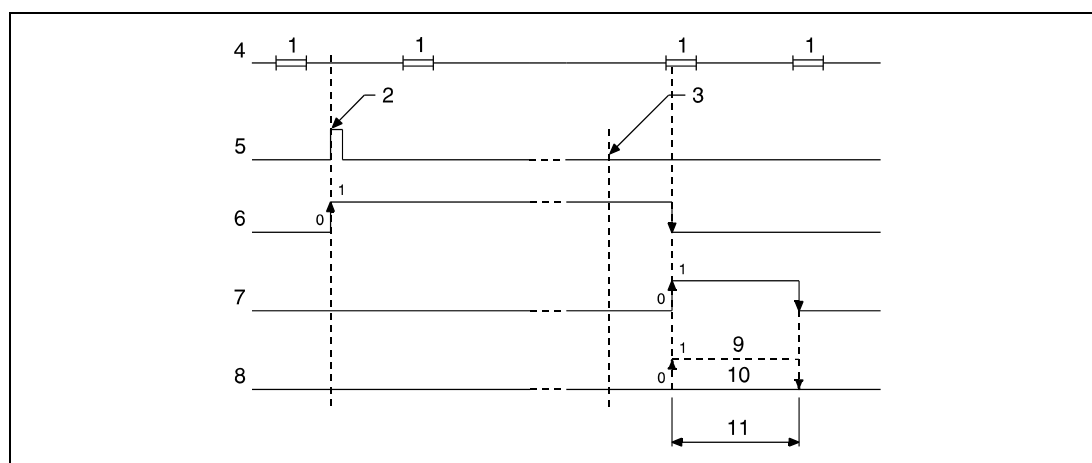
При завершении нормальной (безошибочной) передачи операнд остается сброшенным.

При завершении ненормальной (ошибочной) передачи этот операнд устанавливается при обработке команды END того цикла, в котором была завершена обработка команды RECV. При обработке следующей команды END этот операнд снова сбрасывается.

**ПРИМЕЧАНИЕ** ●<sup>4</sup> В следующей таблице пояснена взаимосвязь между номерами каналов и соответствующими флагами коммуникационных каналов.

Номер канала	1	2	3	4	5	6	7	8
Флаг коммуникационного канала	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

На рисунке ниже изображены операции хост-станции во время выполнения команды RECV.



- 1 обработка команды END
- 2 выполнение команды RECV
- 3 завершение операции
- 4 программа хост-станции
- 5 команда RECV
- 6 флаг коммуникационного канала
- 7 операнд хост-станции, устанавливаемый по окончании операции (d2)
- 8 индикация состояния после завершения операции ((d2) + 1)
- 9 завершение ненормальной (ошибочной) передачи
- 10 завершение нормальной (безошибочной) передачи
- 11 один цикл

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

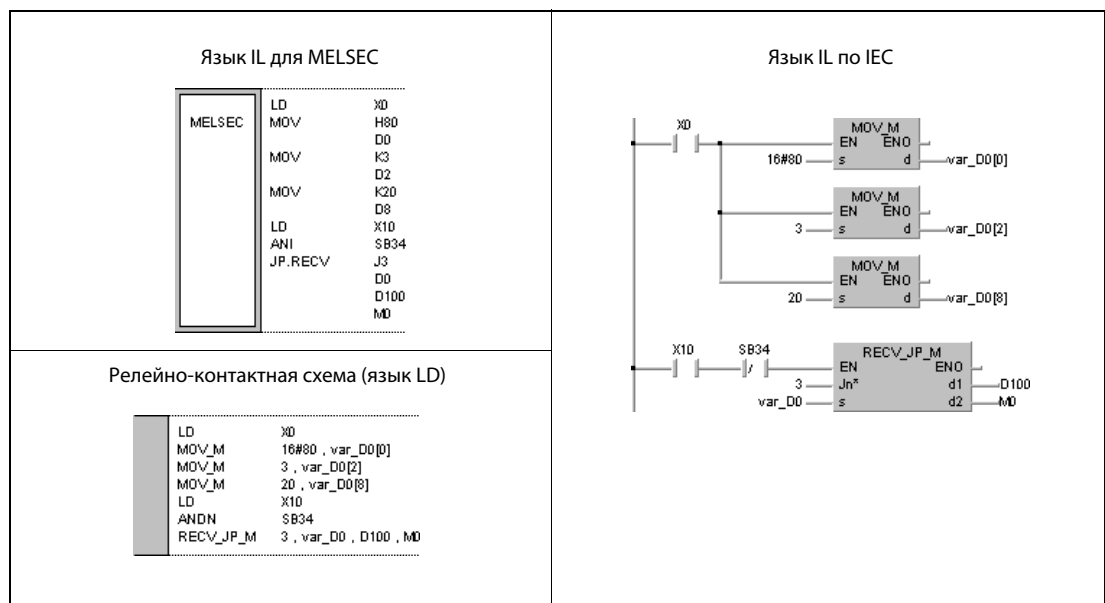
- Содержимое управляющих данных находится вне допустимого диапазона (код ошибки 4100).
- Сеть с номером Jn не соединена со станцией (код ошибки 4102).
- Модуль по адресу ввода-вывода Un не является модулем сетевой коммуникации (код ошибки 2111).

**Пример**

JP.RECV

Следующая программа при положительном фронте сигнала X10 считывает данные станции, переданные с помощью команды SEND. Выполнение команды RECV заблокировано размыкающим контактом флага SB34. Более подробная информация о хост-станции, передающей станции и используемых командах MOV содержится в следующей таблице.

Операнд/команда	Обозначение/функция
Хост-станция	—
Главная сеть (хост)	—
Главный канал (хост)	3
Флаг коммуникационного канала	SB34
Передающая станция	—
Сеть передающей станции	3
Канал передающей станции	3
1-я команда MOV	Установка данных времени
2-я команда MOV	Установка канала хост-станции
3-я команда MOV	Установка времени сторожевого таймера (20 с)



**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию можно найти в разд. 3.5.2 „Адресация массивов и регистров в GX IEC Developer“ этого руководства.

8.6.7 REQ

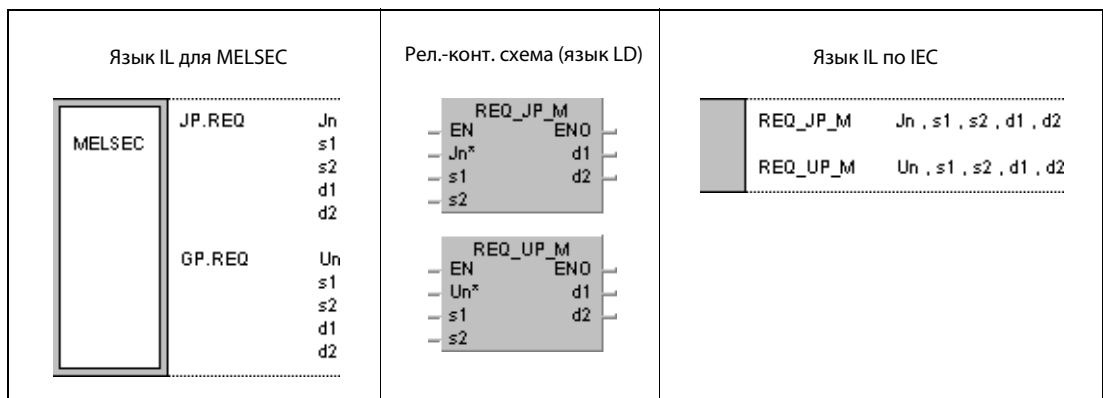
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (сист., польз.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	SM0	8	
s2	—	●	●	—	—	—	—	—			
d1	—	●	●	—	—	—	—	—			
d2	●	●	●	—	—	—	—	—			

GX IEC Developer



Переменные

Операнд	Значение	Тип данных	
		MELSEC	IEC
Jn	Номер сети хост-станции ● <sup>1</sup>		
Un	Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции ● <sup>2</sup>	BIN, 16 бит	ANY16
s1	Первый адрес операнда, в котором сохранены управляющие данные.	адрес	массив [1..18] данных типа ANY16
s2	Первый адрес операнда, в котором сохранены запрашиваемые данные.		массив [1..7] данных типа ANY16
d1	Первый адрес операнда, в котором сохраняются данные ответа.		массив [1..4] данных типа ANY16
d2	Операнд, устанавливаемый после полной обработки команды на один программный цикл.	бит	BOOL

**ПРИМЕЧАНИЯ**

- <sup>1</sup> Номер сети хост-станции должен находиться в диапазоне от 1 до 239. Сеть номер 254 с помощью настроек конфигурируется так, чтобы другие станции могли обращаться к активной станции.
- <sup>2</sup> Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции должен находиться в диапазоне от 0 до Fh. Следует иметь в виду, что в Un компилятор ожидает шестнадцатеричного указания адреса. Десятичный адрес автоматически преобразуется в шестнадцатеричное число.

Выполнение команды REQ возможно только в том случае, если целевая станция построена на процессоре QnA.

При использовании процессора серии "A" в сети MELSECNET/10 команду REQ применять не следует.

В качестве номера целевой станции можно указывать только номера станций с процессорами QnA.

**Обзор операндов управляющих данных**

Операнд	Значение	Функция	Диапазон	Устанавливает
(s1) + 0 Array_s1[1]	Режим выполнения	Подтверждение приема установлено: (бит 0 (b0) = 1, фиксированная настройка)	0001н 0081н	пользователь
	Режим завершения обработки ошибки	Сохранение момента времени, в который была завершена обработка ошибки: – без сохранения данных времени, бит 7 (b7) = 0 – сохранение данных времени, бит 7 (b7) = 1 (данные времени начиная с (s1) + 11 (Array_s1[12]))		
(s1) + 1 Array_s1[2]	Состояние после завершения обработки команды	Индикация состояния по окончании обработки команды: – 0 = ошибок нет (нормальное завершение обработки) – не равно 0: = код ошибки ● <sup>1</sup>	—	система
(s1) + 2 Array_s1[3]	Канал, используемый хост-станцией	Указание канала, который будет применять хост-станция.	от 1 до 8	пользователь
(s1) + 3 Array_s1[4]	Фиктивный операнд	Не используется.	—	—
(s1) + 4 Array_s1[5]	Номер сети целевой станции	Номер сети станции, из которой происходит считывание.	от 1 до 239 254 ● <sup>2</sup>	пользователь
(s1) + 5 Array_s1[6]	Номер целевой станции	Указание номера целевой станции	от 1 до 64	пользователь
(s1) + 6 Array_s1[7]	Фиктивный операнд	Не используется.	—	—
(s1) + 7 Array_s1[8]	Количество попыток передачи	Указание количества попыток, которое требуется предпринять, чтобы полностью обработать команду REQ за контрольное время, заданное в (s1) + 8 (Array_s1[9]).	от 1 до 15	пользователь
	Количество выполненных попыток передачи	Сохранение выполненных попыток передачи.	—	система
(s1) + 8 Array_s1[9]	Настройка сторожевого таймера для контроля времени приема	Контрольное время для операции в секундах. Если операция не завершается за указанное время, передача повторяется количество раз, указанное в (s1) + 7 (Array_s1[8]).	от 1 до 32767 0 = 10 секунд (неизменная настройка)	пользователь
(s1) + 9 Array_s1[10]	Длина запрашиваемых данных	Указание длины запрашиваемых данных. Если данные времени считываются = 2 Если данные времени записываются = 7 Режим RUN/STOP удаленного модуля = 4	2, 7, 4	пользователь
(s1) + 10 Array_s1[11]	Длина данных ответа	Сохранение длины данных ответа. Если данные времени считываются = 2	0, 4	пользователь
(s1) + 11 Array_s1[12]	Флаг сохранения данных времени (устанавливается только при наличии ошибки)	Этот флаг сохраняет состояние опции сохранения данных времени, указанной в (s1) + 0 (Array_s1[1]): без сохранения данных времени = 0 сохранение данных времени = 1	—	система
(s1) + 12 Array_s1[13]	Данные времени (устанавливаются только при наличии ошибки)	старший байт = год (от 0 до 99) младший байт = месяц (от 1 до 12)	—	система
(s1) + 13 Array_s1[14]		старший байт = день (от 1 до 31) младший байт = час (от 0 до 23)		
(s1) + 14 Array_s1[15]		старший байт = минута (от 0 до 59) младший байт = секунда (от 0 до 59)		
(s1) + 15 Array_s1[16]		старший байт = 00н младший байт = день недели (от 0 до 6) (воскресенье = 0, суббота = 6)		

Операнд	Значение	Функция	Диапазон	Устанавливает
(s1) + 16 Array_s1[17]	Номер сети, в которой возникла ошибка	Сохранение номера сети станции, в которой возникла ошибка. Если состояние после завершения выполнения команды гласит "Канал используется (F7C1 н)", сохранение не выполняется. Номер сети находится в диапазоне от 1 до 239.	—	система
(s1) + 17 Array_s1[18]	Номер станции, в которой возникла ошибка	Сохранение номера станции, в которой возникла ошибка. Если состояние после завершения выполнения команды гласит "Канал используется (F7C1н)", сохранение не выполняется. Номер станции находится в диапазоне от 1 до 64.	—	система

- <sup>1</sup> Дополнительная информация о кодах ошибок имеется в руководстве "MELSECNET/10 для сетевых систем QnA".
- <sup>2</sup> Если в Jn указано значение 254, выбран номер сети 254.

**Данные запроса/ответа во время операции записи/считывания данных времени**

**Данные запроса**

Операнд	Значение	Функция	Считывание данных времени	Запись данных времени
(s2) + 0 Array_s2[1]	Режим запроса	0001n = считывание данных времени 0011n = запись данных времени	●	●
(s2) + 1 Array_s2[2]	Режим запроса подпрограммы	0002n = считывание данных времени 0001n = запись данных времени	●	●
(s2) + 2 Array_s2[3]	Схема обновления	Указывает, какие данные времени в операндах с (s2) + 3 (Array_s2[4]) по (s2) + 6 (Array_s2[7]) должны обновляться. Если операнд установлен (1), то соответствующее значение времени или даты обновляется.  b15 ----- b7 b6 b5 b4 b3 b2 b1 b0 0                    0 W Sec Min H D M Y		●
(s2) + 3 Array_s2[4]	Обновляемые данные месяца и года	Месяц и год (последние 2 знака) в двоично-десятичной кодировке  b15 ----- b8 b7 ----- b0 M (01n - 12n)    Y (00n - 99n)		●
(s2) + 4 Array_s2[5]	Обновляемые данные часов и дня	Час и день в двоично-десятичной кодировке  b15 ----- b8 b7 ----- b0 H (00n - 23n)    D (01n - 31n)		●
(s2) + 5 Array_s2[6]	Обновляемые данные минут и секунд	Секунды и минуты в двоично-десятичной кодировке  b15 ----- b8 b7 ----- b0 Sec (00n - 59n)    Min (00n - 59n)		●
(s2) + 6 Array_s2[7]	Обновляемые данные дня недели	День недели в двоично-десятичной кодировке (00n = воскресенье, 06n = суббота)  b15 ----- b8 b7 ----- b0 00n                W (00n - 06n)		●

- M = месяц
- Y = год
- H = час
- D = день
- Sec = секунда
- Min = минута
- W = день недели

## Данные ответа

Операнд	Значение	Функция	Считывание данных времени	Запись данных времени
(d1) + 0 Array_d1[1]	Считанные данные месяца и года	Месяц и год (последние 2 знака) в двоично-десятичной кодировке  b15 ----- b8 b7 ----- b0 M (01н - 12н) Y (00н - 99н)	●	
(d1) + 1 Array_d1[2]	Считанные данные часов и дня	Час и день в двоично-десятичной кодировке  b15 ----- b8 b7 ----- b0 H (00н - 23н) D (01н - 31н)	●	
(d1) + 2 Array_d1[3]	Считанные данные минут и секунд	Секунды и минуты в двоично-десятичной кодировке  b15 ----- b8 b7 ----- b0 Sec (00н - 59н) Min (00н - 59н)	●	
(d1) + 3 Array_d1[4]	Считанное значение дня недели	День недели в двоично-десятичной кодировке (00н = воскресенье, 06н = суббота)  b15 ----- b8 b7 ----- b0 00н W (00н - 06н)	●	

M = месяц  
 Y = год  
 H = час  
 D = день  
 Sec = секунда  
 Min = минута  
 W = день недели

**ПРИМЕЧАНИЕ** Если в модуле центрального процессора целевой станции активирована защита от записи (включен системный выключатель 1, SW5 (QnA, Q4AR) или SW1 (QnAS)), то операции записи и считывания не возможны.



**Данные запроса во время операции RUN/STOP на другой станции**

Операнд	Значение	Функция	Операция RUN	Операция STOP
(s2) + 0 Array-s2[1]	Режим запроса	0010н	●	●
(s2) + 1 Array_s2[2]	Режим запроса подпрограммы	0001н = операция RUN на другой станции 0002н = операция STOP на другой станции	●	●
(s2) + 2 Array_s2[3]	Режим	Указание, должна ли операция RUN на другой станции выполняться принудительно. 0001н = Не выполнять принудительную операцию RUN 0003н = Принудительное выполнение операции RUN (устанавливается во время операции STOP на другой станции) Если станция, выполняющая операцию STOP на другой станции, не может выполнить операцию RUN, операция RUN может быть принудительно выполнена с другой станции.	●	●
(s2) + 3 Array_s2[4]	Режим стирания	Указание состояния памяти центрального процессора во время выполнения операции RUN на другой станции. 0000н = не стирать (устанавливается во время операции STOP на другой станции) 0001н = стирать (кроме областей фиксации) 0002н = стирать (включая области фиксации)	●	●

**ПРИМЕЧАНИЯ** Функция RUN/STOP на другой станции может быть выполнена только в том случае, если выключатель "RUN/STOP" центрального процессора QnA целевой станции находится в положении "RUN".

Если на модуле центрального процессора QnA целевой станции активирована защита от записи (включен системный выключатель 1, SW5 (QnA, Q4AR) или SW1 (QnAS)), то операции записи и считывания не возможны.

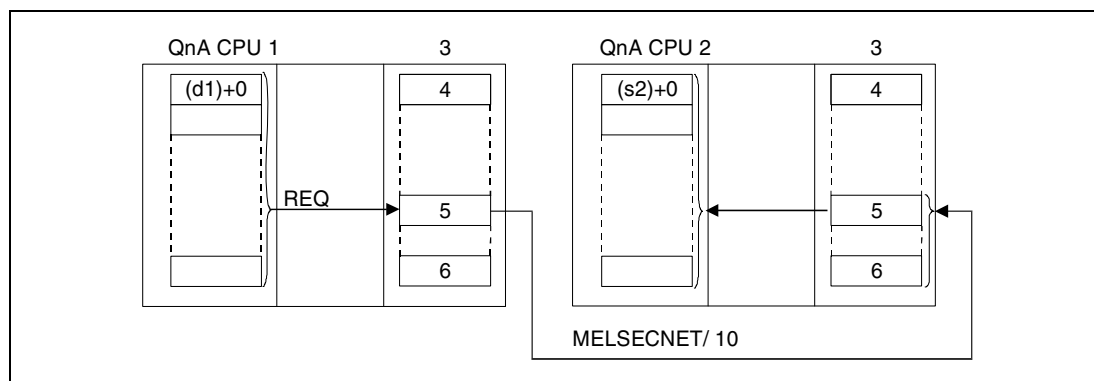
Если целевая станция уже переключена в режим STOP/PAUSE другой станцией, то принудительное выполнение операции RUN возможно только в том случае, если в (s2) + 2 выбран режим "Не выполнять принудительную операцию RUN (0001н)".

При сбросе центрального процессора QnA целевой станции, выполняющей операцию RUN/STOP, информация об операции RUN/STOP в целевой станции утрачивается.

**Принцип действия****Запрос данных от другой станции****REQ      Команда запроса**

Команда REQ передает запрошенные данные, хранящиеся начиная с  $(d1) + 0$  (Array\_d1[1]), в станцию, подключенную к сети MELSECNET/10. Номер станции и номер сети указаны в управляющих данных. Данные сохраняются, начиная с  $(s2) + 0$  (Array\_s2[1]).

По окончании операции передачи запрошенных данных в целевой станции устанавливается операнд, указанный в d2.



- 1 хост-станция
- 2 целевая станция
- 3 модуль сетевой коммуникации
- 4 канал 1
- 5 канал n
- 6 канал 8

С помощью релейной станции и установленных параметров маршрутизации можно обращаться и к станциям в других сетях.

Команды коммуникации не могут выполняться одновременно из нескольких источников с общим доступом к одним и тем же каналам. При одновременном выполнении команд из двух или более источников выполнение других команд коммуникации предотвращается с помощью режима квитирования (handshake) обеих активных станций.

Состояние выполнения и состояние после завершения (нормальное, ненормальное) команды REQ можно контролировать с помощью:

- флага (●<sup>3</sup>) используемого коммуникационного канала,
- операнда, устанавливаемого по окончании операции (d2), и
- индикации состояния после завершения операции (безошибочная или ошибочная передача)  $((d2) + 1)$

Контроль осуществляется следующим образом:

Флаг коммуникационного канала

Этот флаг устанавливается во время выполнения команды REQ. Флаг сбрасывается при обработке команды END того цикла, в котором была завершена операция передачи.

Операнд хост-станции, показывающий завершение операции

Этот операнд устанавливается при обработке команды END того цикла, в котором была завершена операция. При обработке следующей команды END этот операнд снова сбрасывается.

Индикация состояния завершения операции

Этот операнд устанавливается в зависимости от результата после завершения операции.

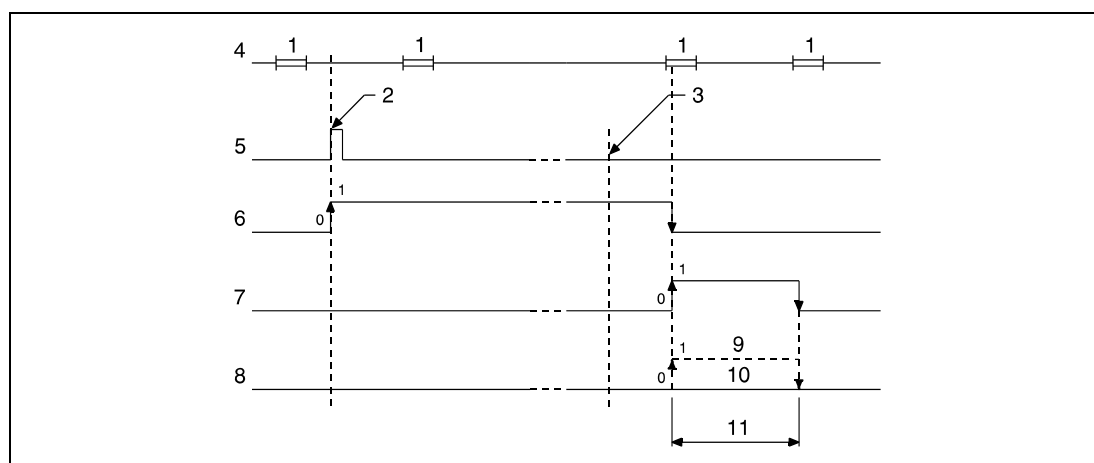
При завершении нормальной (безошибочной) передачи операнд остается сброшенным.

При завершении ненормальной (ошибочной) передачи этот операнд устанавливается при обработке команды END того цикла, в котором была завершена обработка команды REQ. При обработке следующей команды END этот операнд снова сбрасывается.

**ПРИМЕЧАНИЕ** ●<sup>3</sup> В следующей таблице пояснена взаимосвязь между номерами каналов и соответствующими флагами коммуникационных каналов.

Номер канала	1	2	3	4	5	6	7	8
Флаг коммуникационного канала	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

На рисунке ниже изображены операции хост-станции во время выполнения команды REQ.



- 1 обработка команды END
- 2 выполнение команды REQ
- 3 завершение операции
- 4 программа хост-станции
- 5 команда REQ
- 6 флаг коммуникационного канала
- 7 операнд хост-станции, устанавливаемый по окончании операции ((d2)
- 8 индикация состояния после завершения операции ((d2) + 1)
- 9 завершение ненормальной (ошибочной) передачи
- 10 завершение нормальной (безошибочной) передачи
- 11 один цикл

**Источники  
ошибок**

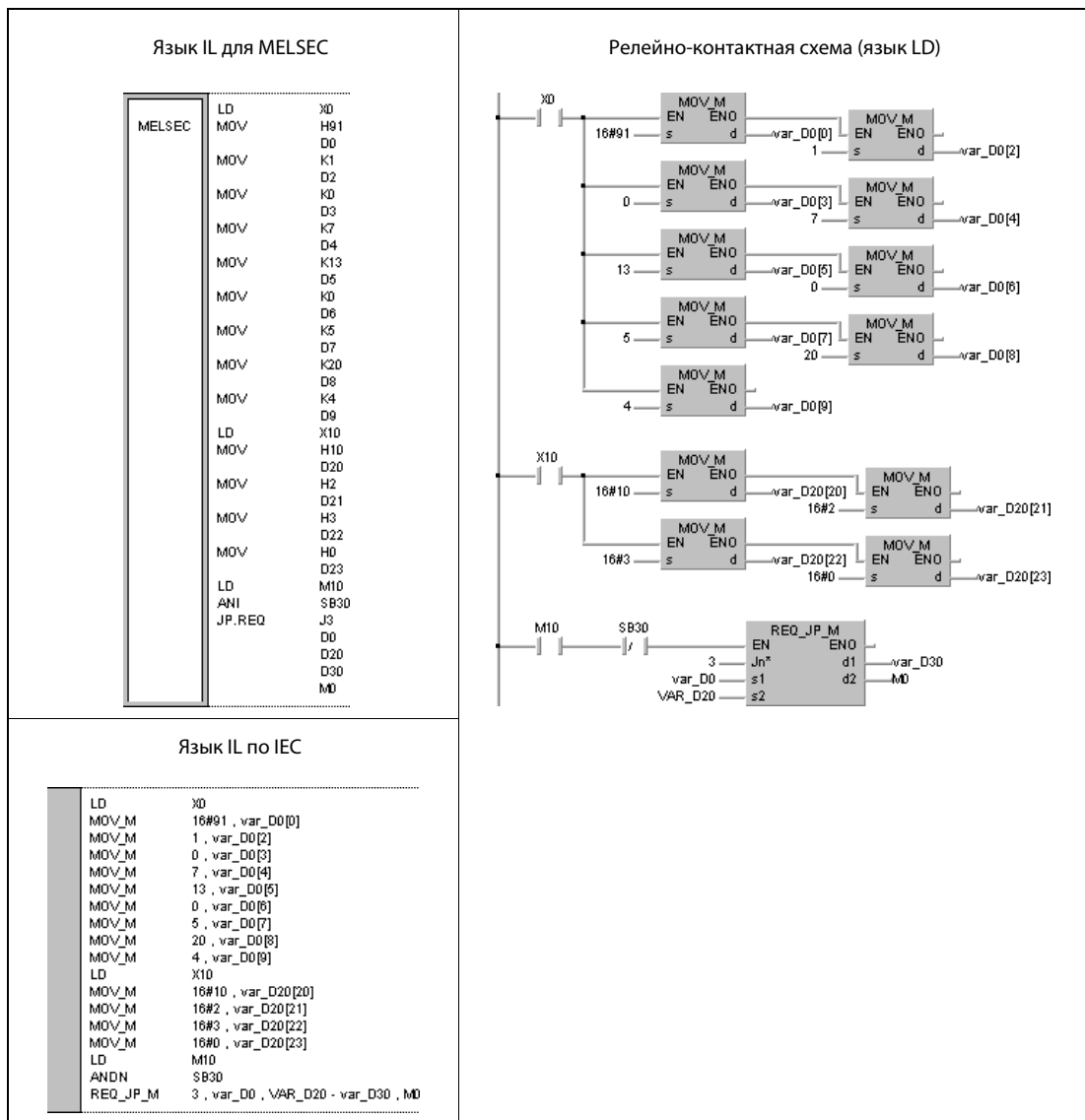
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Содержимое управляющих данных находится вне допустимого диапазона (код ошибки 4100).
- Сеть с номером Jn не соединена со станцией (код ошибки 4102).
- Модуль по адресу ввода-вывода Un не является модулем сетевой коммуникации (код ошибки 2111).

**Пример** JP.REQ

Следующая программа при положительном фронте сигнала X10 выполняет операцию STOP на целевой станции. Выполнение команды REQ заблокировано размыкающим контактом флага SB30. Более подробная информация о хост-станции, целевой станции и применении команд MOV содержится в следующей таблице.

Операнд/команда	Обозначение/функция
Хост-станция	—
Главная сеть (хост)	—
Главный канал (хост)	1
Флаг коммуникационного канала	SB30
Целевая станция	13
Целевая сеть	7
Целевой канал	—
1-я команда MOV	Установка данных времени
2-я команда MOV	Установка канала хост-станции
3-я команда MOV	—
4-я команда MOV	Установка номера сети целевой станции
5-я команда MOV	Установка номера целевой станции
6-я команда MOV	—
7-я команда MOV	Установка количества попыток передачи
8-я команда MOV	Установка времени сторожевого таймера (20 с)
9-я команда MOV	Установка количества передаваемых блоков (4)
10-я команда MOV	Установка режима запроса
11-я команда MOV	Установка режима запроса подпрограммы
12-я команда MOV	Установка режима
13-я команда MOV	Установка режима стирания



**ПРИМЕЧАНИЕ** Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 8.6.8 ZNFR

Процессор

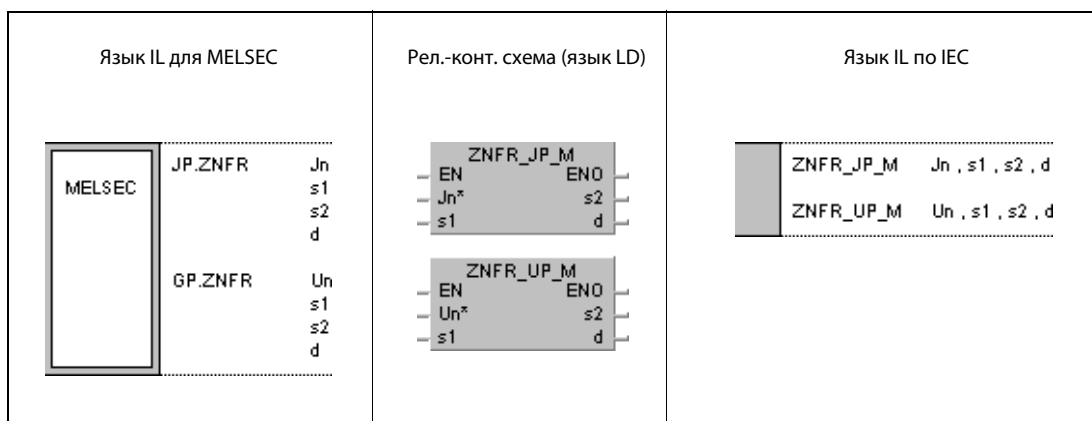
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (сист., польз.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	—	—	8
s2	—	● <sup>1</sup>	—	—	—	—	—	—	—		
d	●	—	—	—	—	—	—	—	—		

<sup>1</sup> только регистры связи

GX IEC Developer



Переменные

Операнд	Значение	Тип данных	
		MELSEC	IEC
Jn	Номер сети хост-станции ● <sup>1</sup>		
Un	Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции ● <sup>2</sup>	BIN, 16 бит	ANY16
s1	Первый адрес операнда, в котором сохранены управляющие данные.	адрес	массив [1..15] данных типа ANY16
s2	Первый адрес регистра связи (W) в хост-станции, в котором сохраняются считанные данные.		ANY16
d	Операнд, устанавливаемый после полной обработки команды на один программный цикл.	бит	BOOL

ПРИМЕЧАНИЯ

- <sup>1</sup> Номер сети хост-станции должен находиться в диапазоне от 1 до 239. Сеть номер 254 с помощью настроек конфигурируется так, чтобы другие станции могли обращаться к активной станции.
- <sup>2</sup> Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции должен находиться в диапазоне от 0 до FEn. Следует иметь в виду, что в Un компилятор ожидает шестнадцатеричного указания адреса. Десятичный адрес автоматически преобразуется в шестнадцатеричное число.

## Обзор операндов управляющих данных

Операнд	Значение	Функция	Диапазон	Устанавливает
(s1) + 0 Array_s1[1]	Режим выполнения	Подтверждение приема установлено: (бит 0 (b0) = 1, фиксированная настройка)	0001h 0081h	пользователь
	Режим завершения обработки ошибки	Сохранение момента времени, в который была завершена обработка ошибки: – без сохранения данных времени, бит 7 (b7) = 0 – сохранение данных времени, бит 7 (b7) = 1 (данные времени начиная с (s1) + 11 (Array_s1[12]))		
(s1) + 1 Array_s1[2]	Состояние после завершения обработки команды	Индикация состояния по окончании обработки команды: – 0 = ошибок нет (нормальное завершение обработки) – не равно 0: = код ошибки ● <sup>3</sup>	—	система
(s1) + 2 Array_s1[3]	Фиктивный операнд	Не используется.	—	—
(s1) + 3 Array_s1[4]	Адрес буферной памяти	Указание первого адреса буферной памяти	● <sup>4</sup>	пользователь
(s1) + 4 Array_s1[5]	Фиктивный операнд	Не используется.	—	—
(s1) + 5 Array_s1[6]	Номер целевой станции	Указание номера удаленной станции ввода-вывода, которая будет считывать данные.	от 1 до 64	пользователь
(s1) + 6 Array_s1[7]	Местоположение специального модуля в ряду модулей	Указание позиции, в которой расположен специальный модуль в ряду специальных модулей в целевой станции.	—	пользователь
(s1) + 7 Array_s1[8]	Фиктивный операнд	Не используется.	—	—
(s1) + 8 Array_s1[9]	Фиктивный операнд	Не используется.	—	—
(s1) + 9 Array_s1[10]	Длина считываемых данных	Указание количества считываемых данных	от 1 до 256	пользователь
(s1) + 10 Array_s1[11]	Флаг сохранения данных времени (устанавливается только при наличии ошибки)	Этот флаг сохраняет состояние опции сохранения данных времени, указанной в (s1) + 0 (Array_s1[1]): без сохранения данных времени = 0 сохранение данных времени = 1	—	система
(s1) + 11 Array_s1[12]	Данные времени (устанавливаются только при наличии ошибки)	старший байт = год (от 0 до 99) младший байт = месяц (от 1 до 12)	—	система
(s1) + 12 Array_s1[13]		старший байт = день (от 1 до 31) младший байт = час (от 0 до 23)		
(s1) + 13 Array_s1[14]		старший байт = минута (от 0 до 59) младший байт = секунда (от 0 до 59)		
(s1) + 14 Array_s1[15]		старший байт = 00h младший байт = день недели (от 0 до 6) (воскресенье = 0, суббота = 6)		

●<sup>3</sup> Дополнительная информация о кодах ошибок имеется в руководстве "MELSECNET/10 для сетевых систем QnA".

●<sup>4</sup> Дополнительная информация имеется в руководстве по специальному модулю, выполняющему операцию считывания.

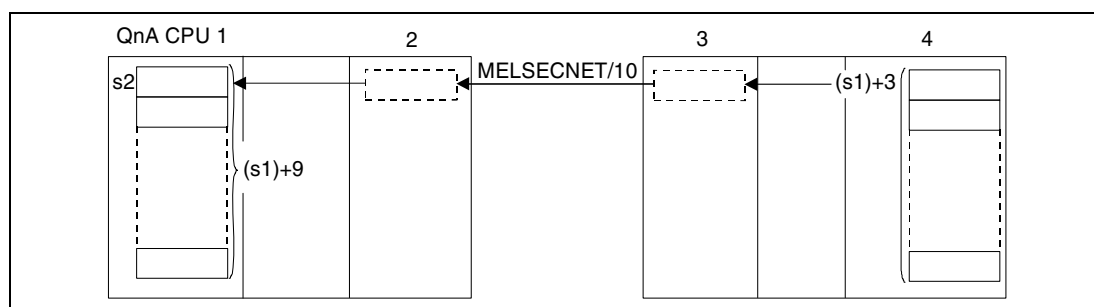


**Принцип действия**

**Считывание данных из специальных модулей в удаленных станциях ввода-вывода ZNFR Команда считывания**

Команда ZNFR считывает данные, сохраненные в буферной памяти специального модуля в удаленной станции ввода-вывода, подключенной к сети MELSECNET/10. Станция ввода-вывода указана в управляющих данных. Считанные из модуля данные сохраняются в хост-станции, начиная с s2.

По окончании операции считывания в удаленной станции ввода-вывода устанавливается операнд, указанный в d.



- 1 главная станция (хост/мастер)
- 2 модуль сетевой коммуникации (хост/мастер)
- 3 удаленная станция ввода-вывода (целевая станция)
- 4 специальный модуль (целевая станция/удаленная станция ввода-вывода)

Операцию считывания из удаленной станции ввода-вывода можно выполнить только через модуль сетевой коммуникации, который подключен к той же сети, что и удаленная станция ввода-вывода, подключенная к сети MELSECNET/10.

Команда ZNFR не может выполняться одновременно из нескольких источников одного и того же специального модуля. При одновременном выполнении команд из двух или более источников выполнение других команд ZNFR предотвращается благодаря квитированию связи (handshake) между двумя активными станциями.

Блокировочный сигнал, передаваемый во время выполнения команды ZNFR, содержит

- сигналы запроса считывания/записи,
- сигналы завершения операций считывания/записи,
- операнд, устанавливаемый по окончании операции считывания (d)
- и индикацию состояния после завершения операции (безошибочная или ошибочная передача) (d + 1). Эти сигналы и операнды описаны ниже:

**Сигналы запроса считывания/записи**

Этот сигнал устанавливается при выполнении специализированных команд коммуникации серии QnA. Сигнал сбрасывается при обработке команды END того цикла, в котором были завершены операции считывания/записи.

**Сигналы завершения операций считывания/записи**

Этот сигнал сбрасывается при выполнении специализированных команд коммуникации серии QnA. Сигнал устанавливается при обработке команды END того цикла, в котором были завершены операции считывания/записи.

**Операнд хост-станции, показывающий завершение операции**

Этот операнд устанавливается при обработке команды END того цикла, в котором была завершена обработка команды ZNFR. При обработке следующей команды END этот операнд снова сбрасывается.

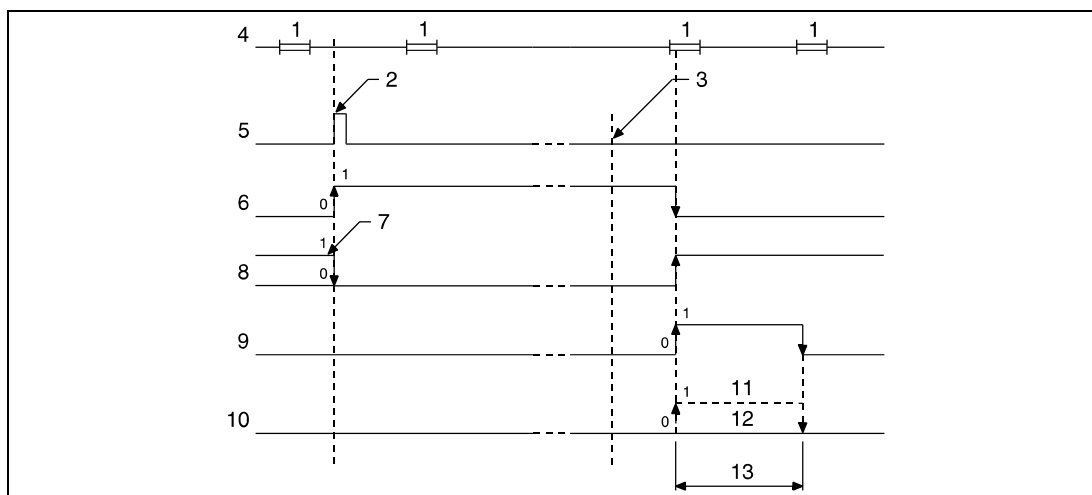
Индикация состояния завершения операции

Этот операнд устанавливается в зависимости от результата после завершения обработки команды ZNFR.

При завершении нормальной (безошибочной) передачи операнд остается сброшенным.

После завершения ненормальной передачи (т. е. передачи, содержащей ошибки) этот операнд устанавливается при обработке команды END того цикла, в котором была завершена обработка команды ZNFR. При обработке следующей команды END этот операнд снова сбрасывается.

На рисунке ниже изображены операции хост-станции во время выполнения команды ZNFR.



- <sup>1</sup> обработка команды END
- <sup>2</sup> выполнение команды ZNFR
- <sup>3</sup> завершение операции
- <sup>4</sup> программа хост-станции
- <sup>5</sup> команда ZNFR
- <sup>6</sup> сигнал запроса считывания/записи
- <sup>7</sup> после выполнения специализированных команд коммуникации
- <sup>8</sup> завершение операции считывания/записи
- <sup>9</sup> операнд хост-станции, устанавливаемый по окончании операции (d)
- <sup>10</sup> индикация состояния после завершения операции (d + 1)
- <sup>11</sup> завершение ненормальной (ошибочной) передачи
- <sup>12</sup> завершение нормальной (безошибочной) передачи
- <sup>13</sup> один цикл

Регистры связи в s2 устанавливаются параметрами сетевой коммуникации "M ← R (к мастер-станции от удаленной станции ввода-вывода)". Они находятся в области, установленной в параметрах обновления канала.

Для выполнения команды ZNFR необходимы маркеры связи и регистры связи, которые может использовать операционная система. Ниже указано количество маркеров связи и регистров связи, используемых операционной системой для соответствующего специального модуля:

Для  $M \rightarrow R$  (от мастер-станции к удаленной станции ввода-вывода):  
маркеров связи = 4, регистров связи = 4

Для  $M \leftarrow R$  (к мастер-станции от удаленной станции ввода-вывода):  
маркеров связи = 4, регистров связи = 4

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

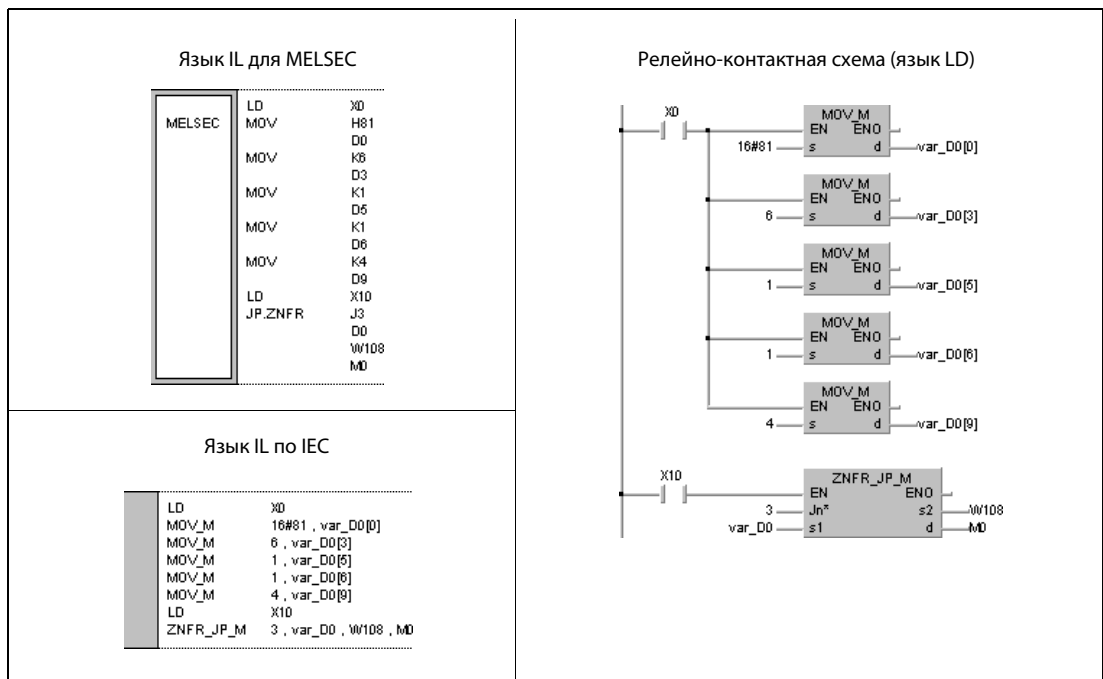
- Содержимое управляющих данных находится вне допустимого диапазона (код ошибки 4100).
- Сеть с номером Jn не соединена со станцией (код ошибки 4102).
- Модуль по адресу ввода-вывода Un не является модулем сетевой коммуникации (код ошибки 2111).

**Пример**

JP.ZNFR

Следующая программа при положительном фронте сигнала X10 считывает данные из адресов 6...9 буферной памяти специального модуля в станции ввода-вывода. Считанные данные сохраняются в регистрах связи с W108 по W10B. Более подробная информация о станции ввода-вывода и использовании команд MOV содержится в следующей таблице.

Операнд/команда	Обозначение/функция
Станция ввода-вывода	1R1
Сеть станций ввода-вывода	3
Специальный модуль	1
1-я команда MOV	Установка данных времени
2-я команда MOV	Установка первого адреса буферной памяти (6)
3-я команда MOV	Установка номера станции ввода-вывода
4-я команда MOV	Установка местоположения специального модуля в ряду модулей
5-я команда MOV	Установка длины считываемых данных



**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 8.6.9 ZNTO

Процессор

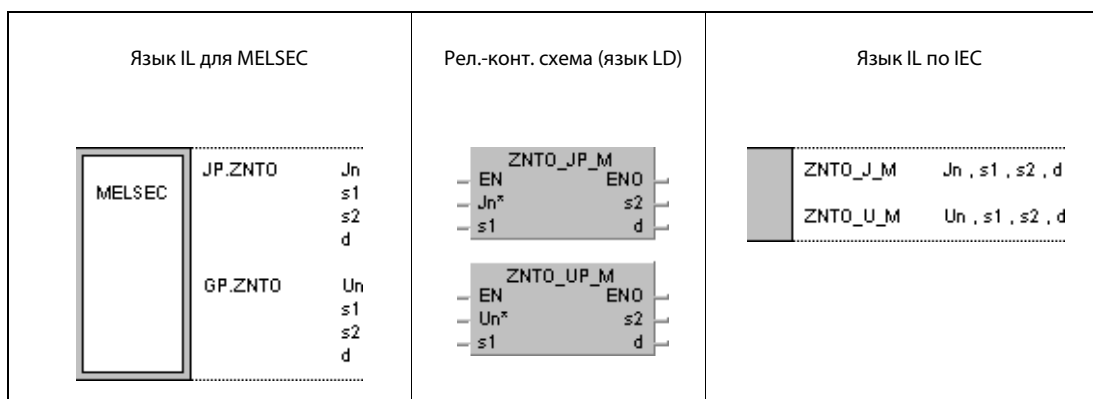
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (сист., польз.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	—	—	8
s2	—	● <sup>1</sup>	—	—	—	—	—	—	—		
d	●	—	—	—	—	—	—	—	—		

<sup>1</sup> только регистры связи

GX IEC Developer



Переменные

Операнд	Значение	Тип данных	
		MELSEC	IEC
Jn	Номер сети хост-станции ● <sup>1</sup>		
Un	Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции ● <sup>2</sup>	BIN, 16 бит	ANY16
s1	Первый адрес операнда, в котором сохранены управляющие данные.	адрес	массив [1..16] данных типа ANY16
s2	Первый адрес регистров связи (W) в хост-станции, в которых сохранены записываемые данные.		ANY16
d	Операнд, устанавливаемый после полной обработки команды на один программный цикл.	бит	BOOL

ПРИМЕЧАНИЯ

- <sup>1</sup> Номер сети хост-станции должен находиться в диапазоне от 1 до 239. Сеть номер 254 с помощью настроек конфигурируется так, чтобы другие станции могли обращаться к активной станции.
- <sup>2</sup> Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции должен находиться в диапазоне от 0 до FEn. Следует иметь в виду, что в Un компилятор ожидает шестнадцатеричного указания адреса. Десятичный адрес автоматически преобразуется в шестнадцатеричное число.

**Обзор операндов управляющих данных**

Операнд	Значение	Функция	Диапазон	Устанавливает
(s1) + 0 Array_s1[1]	Режим выполнения	Подтверждение приема установлено: (бит 0 (b0) = 1, фиксированная настройка)	0001н 0081н	пользователь
	Режим завершения обработки ошибки	Сохранение момента времени, в который была завершена обработка ошибки: – без сохранения данных времени, бит 7 (b7) = 0 – сохранение данных времени, бит 7 (b7) = 1 (данные времени начиная с (s1) + 11 (Array_s1[12]))		
(s1) + 1 Array_s1[2]	Состояние после завершения обработки команды	Индикация состояния по окончании обработки команды: – 0 = ошибок нет (нормальное завершение обработки) – не равно 0: = код ошибки <sup>3</sup>	—	система
(s1) + 2 Array_s1[3]	Фиктивный операнд	Не используется.	—	—
(s1) + 3 Array_s1[4]	Адрес буферной памяти	Указание первого адреса буферной памяти	● <sup>4</sup>	пользователь
(s1) + 4 Array_s1[5]	Фиктивный операнд	Не используется.	—	—
(s1) + 5 Array_s1[6]	Номер целевой станции	Указание номера целевой станции	от 1 до 64	пользователь
(s1) + 6 Array_s1[7]	Номер специального модуля	Указание порядкового номера соответствующего специального модуля в ряду специальных модулей в целевой станции.	—	пользователь
(s1) + 7 Array_s1[8]	Фиктивный операнд	Не используется.	—	—
(s1) + 8 Array_s1[9]	Фиктивный операнд	Не используется.	—	—
(s1) + 9 Array_s1[10]	Длина записываемых данных	Указание количества записываемых данных	от 1 до 256	пользователь
(s1) + 10 Array_s1[11]	Фиктивный операнд	Не используется.	—	—
(s1) + 11 Array_s1[12]	Флаг сохранения данных времени (устанавливается только при наличии ошибки)	Этот флаг сохраняет состояние опции сохранения данных времени, указанной в (s1) + 0 (Array_s1[1]): без сохранения данных времени = 0 сохранение данных времени = 1	—	система
(s1) + 12 Array_s1[13]	Данные времени (устанавливаются только при наличии ошибки)	старший байт = год (от 0 до 99) младший байт = месяц (от 1 до 12)	—	система
(s1) + 13 Array_s1[14]		старший байт = день (от 1 до 31) младший байт = час (от 0 до 23)		
(s1) + 14 Array_s1[15]		старший байт = минута (от 0 до 59) младший байт = секунда (от 0 до 59)		
(s1) + 15 Array_s1[16]		старший байт = 00н младший байт = день недели (от 0 до 6) (воскресенье = 0, суббота = 6)		

●<sup>3</sup> Дополнительная информация о кодах ошибок имеется в руководстве "MELSECNET/10 для сетевых систем QnA".

●<sup>4</sup> Дополнительная информация имеется в руководстве по специальному модулю, выполняющему операцию записи.

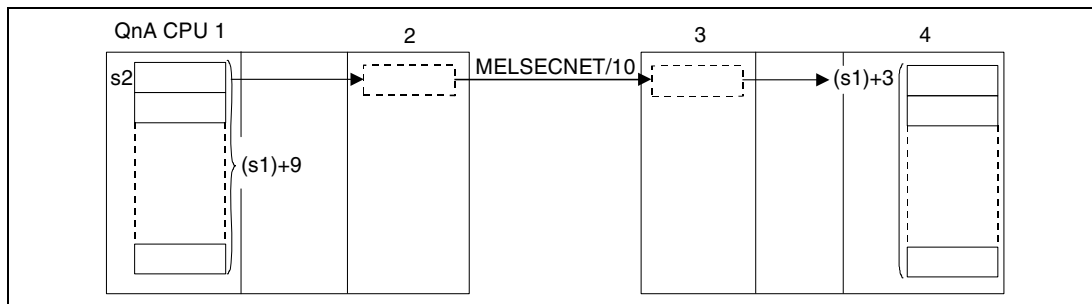
**Принцип действия**

**Запись данных в специальные модули удаленных станций ввода-вывода**

**ZNTO Команда записи**

Команда ZNTO записывает данные хост-станции, хранящиеся, начиная с s2, в буферную память специального модуля удаленной станции ввода-вывода, подключенной к сети MELSECNET/10. Станция ввода-вывода указана в управляющих данных.

По окончании операции записи в удаленной станции ввода-вывода устанавливается операнд, указанный в d.



- 1 главная станция (хост/мастер)
- 2 модуль сетевой коммуникации (хост/мастер)
- 3 удаленная станция ввода-вывода (целевая станция)
- 4 специальный модуль (целевая станция/удаленная станция ввода-вывода)

Операцию записи может выполнить только мастер-станция, подключенная к сети MELSECNET/10, в удаленную станцию ввода-вывода, подключенную к той же сети.

Выполнение команды ZNTO одновременно из нескольких источников в одном и том же специальном модуле не возможно. При одновременном выполнении команд из двух или более источников выполнение других команд ZNTO предотвращается благодаря квитированию связи (handshake) между двумя активными станциями.

Блокировочный сигнал, передаваемый во время выполнения команды ZNTO, содержит

- сигналы запроса считывания/записи,
- сигналы завершения операций считывания/записи,
- операнд, устанавливаемый по окончании операции записи (d),
- и индикацию состояния после завершения операции (безошибочная или ошибочная передача) (d + 1). Эти сигналы и операнд описаны ниже:

**Сигналы запроса считывания/записи**

Этот сигнал устанавливается при выполнении специализированных команд коммуникации серии QnA. Сигнал сбрасывается при обработке команды END того цикла, в котором были завершены операции считывания/записи.

**Сигналы завершения операций считывания/записи**

Этот сигнал сбрасывается при выполнении специализированных команд коммуникации серии QnA. Сигнал устанавливается при обработке команды END того цикла, в котором были завершены операции считывания/записи.

**Операнд хост-станции, показывающий завершение операции**

Этот операнд устанавливается при обработке команды END того цикла, в котором была завершена обработка команды ZNTO. При обработке следующей команды END этот операнд снова сбрасывается.

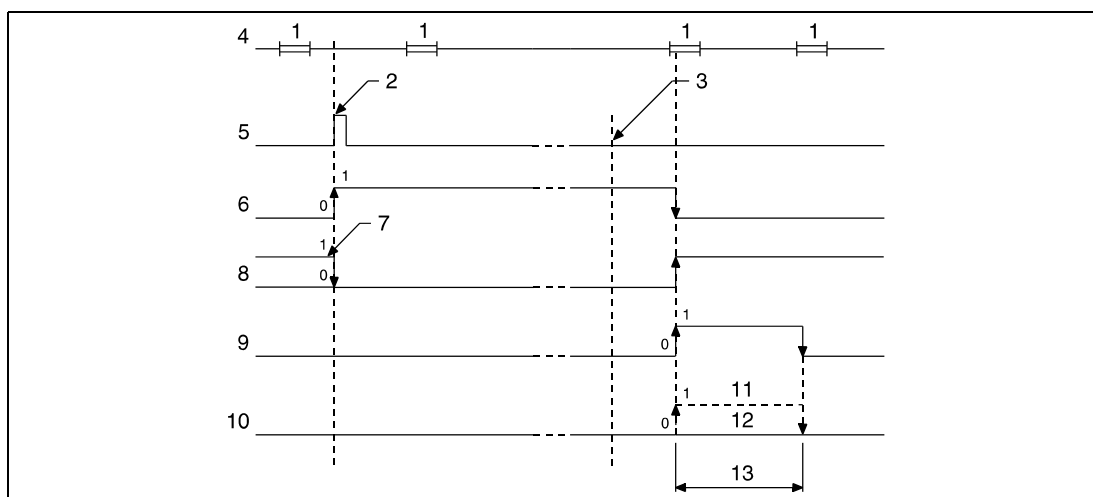
## Индикация состояния завершения операции

Этот операнд устанавливается в зависимости от результата после завершения обработки команды ZNTO.

При завершении нормальной (безошибочной) передачи операнд остается сброшенным.

После завершения ненормальной передачи (т. е. передачи, содержащей ошибки) этот операнд устанавливается при обработке команды END того цикла, в котором была завершена обработка команды ZNTO. При обработке следующей команды END этот операнд снова сбрасывается.

На рисунке ниже изображены операции хост-станции во время выполнения команды ZNTO.



- 1 обработка команды END
- 2 выполнение команды ZNTO
- 3 завершение операции
- 4 программа хост-станции
- 5 команда ZNTO
- 6 сигнал запроса считывания/записи
- 7 после выполнения специализированных команд коммуникации
- 8 завершение операции считывания/записи
- 9 операнд хост-станции, устанавливаемый по окончании операции (d)
- 10 индикация состояния после завершения операции (d + 1)
- 11 завершение ненормальной (ошибочной) передачи
- 12 завершение нормальной (безошибочной) передачи
- 13 один цикл

Регистры связи в s2 устанавливаются параметрами сетевой коммуникации "M ← R (к мастер-станции от удаленной станции ввода-вывода)". Они находятся в области, установленной в параметрах обновления канала.



Для выполнения команды ZNTO необходимы маркеры связи и регистры связи, которые может использовать операционная система. Ниже указано количество маркеров связи и регистров связи, используемых операционной системой для соответствующего специального модуля:

Для  $M \rightarrow R$  (от мастер-станции к удаленной станции ввода-вывода):  
маркеров связи = 4, регистров связи = 4

Для  $M \leftarrow R$  (к мастер-станции от удаленной станции ввода-вывода):  
маркеров связи = 4, регистров связи = 4

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

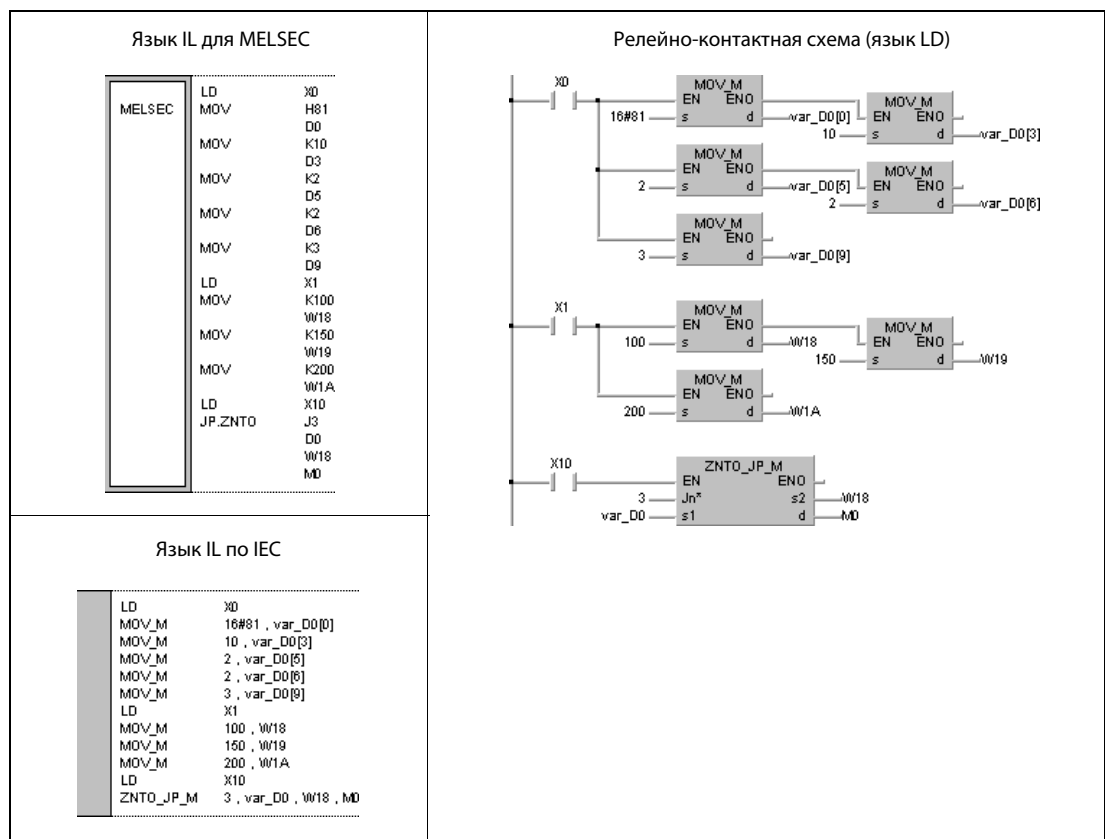
- Содержимое управляющих данных находится вне допустимого диапазона (код ошибки 4100).
- Сеть с номером Jn не соединена со станцией (код ошибки 4102).
- Модуль по адресу ввода-вывода Un не является модулем сетевой коммуникации (код ошибки 2111).

**Пример**

JP.ZNTO

Следующая программа при положительном фронте сигнала X10 записывает данные в адреса 10...12 буферной памяти специального модуля в станции ввода-вывода. Записываемые данные хранятся в регистрах связи W18...W1A хост-станции. Более подробная информация о станции ввода-вывода и использовании команд MOV содержится в следующей таблице.

Операнд/команда	Обозначение/функция
Станция ввода-вывода	1R2
Сеть станций ввода-вывода	3
Специальный модуль	2
1-я команда MOV	Установка данных времени
2-я команда MOV	Установка первого адреса буферной памяти (10)
3-я команда MOV	Установка номера станции ввода-вывода
4-я команда MOV	Установка местоположения специального модуля в ряду модулей
5-я команда MOV	Установка длины записываемых данных
6-я команда MOV	Запись данных в регистры связи с W18 по W1A
7-я команда MOV	
8-я команда MOV	



**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

## 8.7 Команды коммуникации, совместимые с серией "А"

Эти команды позволяют осуществлять обмен данными по сетям MELSECNET и MELSECNET/10 между станциями, работающими на процессорах QnA, между станциями с разными процессорами (QnA и "А"), а также между станциями на процессорах QnA или "А" и удаленными станциями ввода-вывода. Следующая таблица содержит обзор команд.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе МЭК (IEC)	Целевая станция в MELSECNET/10			MELSECNET
			Процессор QnA	Процессор "А"	Удаленные станции ввода-вывода	
Считывание данных QnA из целевых станций в целевых сетях	J.ZNRD	ZNRD_J_M	●	●	—	—
	JP.ZNRD	ZNRD_JP_M				
Считывание данных из локальных станций (возможно только на мастер-станциях)	J.ZNRD	ZNRD_J_M	—	—	—	●
	JP.ZNRD	ZNRD_JP_M				
Запись данных QnA в целевые станции в целевых сетях	J.ZNWR	ZNWR_J_M	●	●	—	—
	JP.ZNWR	ZNWR_JP_M				
Запись данных в локальные станции (возможно только на главных мастер-станциях)	J.ZNWR	ZNWR_J_M	—	—	—	●
	JP.ZNWR	ZNWR_JP_M				
Только серия "А": Считывание данных из локальных станций (возможно только на главных мастер-станциях)	LRDP	LRDP_M	—	—	—	●
		LRDP_MD				
		LRDP_P_MD				
Только серия "А": Запись данных в локальные станции (возможно только на главных мастер-станциях)	LWTP	LWTP_M	—	—	—	●
		LWTP_MD				
		LWTP_P_MD				
Считывание данных из специальных модулей в удаленных станциях ввода-вывода.	RFRP/ G.RFRP	RFRP_U_M	—	—	—	●
		RFRP_UP_M				
Запись данных в специальные модули в удаленных станциях ввода-вывода.	RTOP G.RTOP	RTOP_U_M	—	—	—	●
		RTOP_UP_M				

### 8.7.1 ZNRD

**Процессор**

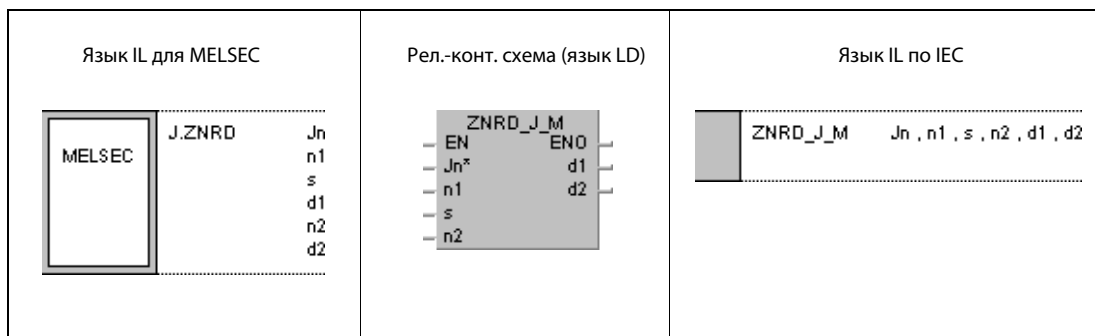
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (сист., польз.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
n1	●	●	●	—	—	—	—	●	—	SM0	32
s	—	● <sup>1</sup>	—	—	—	—	—	—	—		
d1	—	●	●	—	—	—	—	—	—		
n2	●	●	●	—	—	—	—	●	—		
d2	●	●	●	—	—	—	—	—	—		

<sup>1</sup> только T, C, D, и W

**GX IEC Developer**



**Переменные**

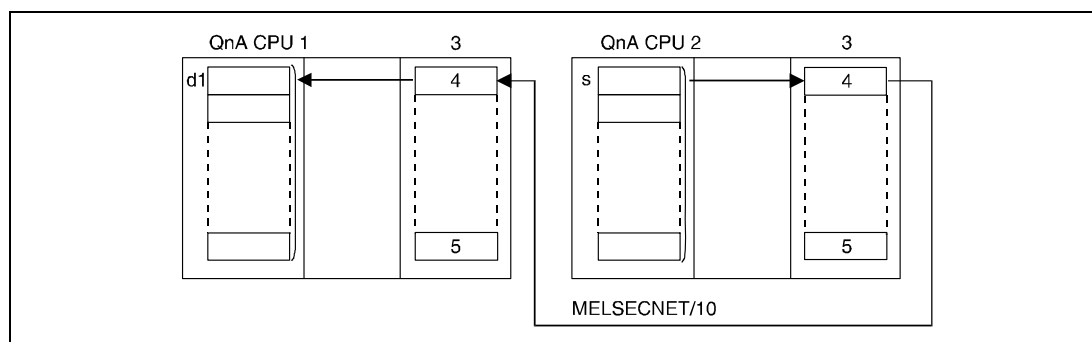
Операнд	Значение	Тип данных
Jn	Номер сети хост-станции ● <sup>1</sup>	BIN, 16 бит
n1	Номер целевой станции	
s	Первый операнд станции, в котором сохранены считываемые данные.	
d1	Первый операнд хост-станции, в котором сохраняются считанные данные.	
n2	Длина принимаемых данных	
d2	Операнд, устанавливаемый после полной обработки команд на один программный цикл.	бит

**ПРИМЕЧАНИЕ** ●<sup>1</sup> Если эта команда используется в MELSECNET(I/II/B), следует указать J0.

**Принцип действия**      **Считывание данных других станций**  
**ZNRD      Команда считывания**

Команда ZNRD считывает указанное в n2 количество слов данных, хранящихся, начиная с s в целевой станции, подключенной к сети MELSECNET/10. Номер целевой станции указан в n1. Номер сети указан в Jn. Считанные данные сохраняются в хост-станции, начиная с d1.

По окончании операции считывания в целевой станции устанавливается операнд, указанный в d2.



- 1 хост-станция
- 2 целевая станция
- 3 модуль сетевой коммуникации
- 4 канал 1
- 5 канал 8

Операция считывания возможна только в отношении целевой станции, подключенной к той же сети MELSECNET/10, что и хост-станция.

Операция считывания с локальной станции возможна только в отношении мастер-станции, подключенной к сети MELSECNET.

Номера сетей в Jn можно указывать в диапазоне от 1 до 239. Указание номера сети 0 (J0) равносильно указанию MELSECNET в качестве системы сетевой коммуникации.

В системе MELSECNET номер целевой сети (Jn) имеет фиксированную настройку 0 (J0). Номера целевых сетей (Jn) от 1 до 239 используются в MELSECNET/10.

В n1 можно указать номер станции между 1 и 64.

При использовании сетевой системы MELSECNET/B номер станции может находиться в диапазоне от 1 до 31.

Указываемая в n2 длина принимаемых данных (количество слов данных) может быть в диапазоне от 1 до 230.

Команда ZNRD может выполняться как на станциях, оснащенных процессорами AnU, так и на станциях, оснащенных процессорами QnA.

Команды коммуникации не могут выполняться одновременно из нескольких источников с общим доступом к одним и тем же каналам. При одновременном выполнении команд из двух или более источников выполнение других команд коммуникации предотвращается с помощью режима квитирования (handshake) обеих активных станций.

Для выполнения команды ZNRD обе станции – главная (хост) и целевая – используют канал 1 модуля сетевой коммуникации. При использовании нескольких команд ZNRD происходит многократное обращение к каналу 1, хотя канал 1 модуля сетевой коммуникации может быть одновременно использован одной командой только один раз. Чтобы предотвратить одновременное выполнение нескольких команд, следует предусмотреть блокировку на основе сигнала запроса считывания/записи и операнда, устанавливаемого после завершения операции.

Выполнение и конец выполнения команды ZNRD отображаются с помощью флага коммуникационного канала (SB30) и операнда (d2), устанавливаемого при завершении операции, следующим образом:

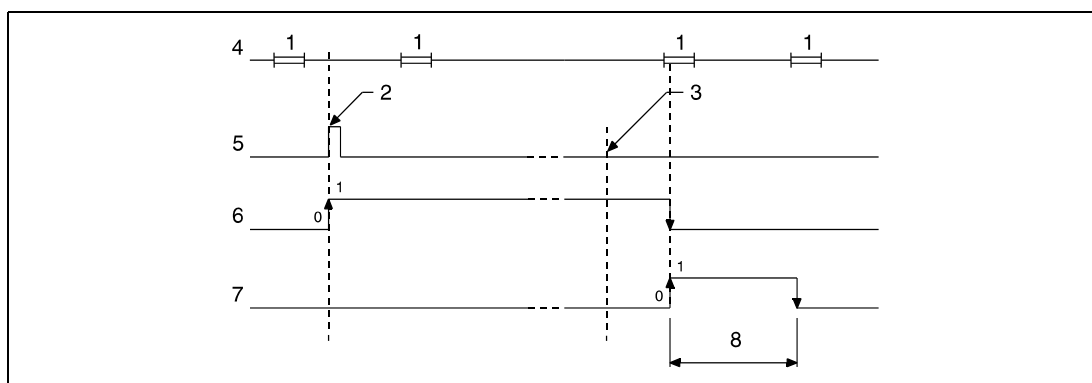
Флаг коммуникационного канала

Этот флаг устанавливается во время выполнения команды ZNRD. Флаг сбрасывается при обработке команды END того цикла, в котором была завершена операция считывания.

Операнд хост-станции, показывающий завершение операции считывания

Этот операнд устанавливается при обработке команды END того цикла, в котором завершена операция считывания. При обработке следующей команды END этот операнд снова сбрасывается.

На рисунке ниже изображены операции хост-станции во время выполнения команды ZNRD.



- 1 обработка команды END
- 2 выполнение команды ZNRD
- 3 завершение операции
- 4 программа хост-станции
- 5 команда ZNRD
- 6 флаг коммуникационного канала (SB30)
- 7 операнд хост-станции, устанавливаемый по окончании операции (d2)
- 8 один цикл

Состояние после завершения команды ZNRD (нормальное, не нормальное) отображается с помощью регистра завершения обработки команды ZNRD (SW31) следующим образом:

При нормальном (безошибочном) завершении операции содержимое регистра SW31 равно 0.

При ненормальном (ошибочном) завершении операции в регистре SW31 сохраняется соответствующий код ошибки.

**ПРИМЕЧАНИЕ** *Дополнительная информация о кодах ошибок имеется в руководстве "MELSECNET/10 для сетевых систем QnA".*

**Источники ошибок**

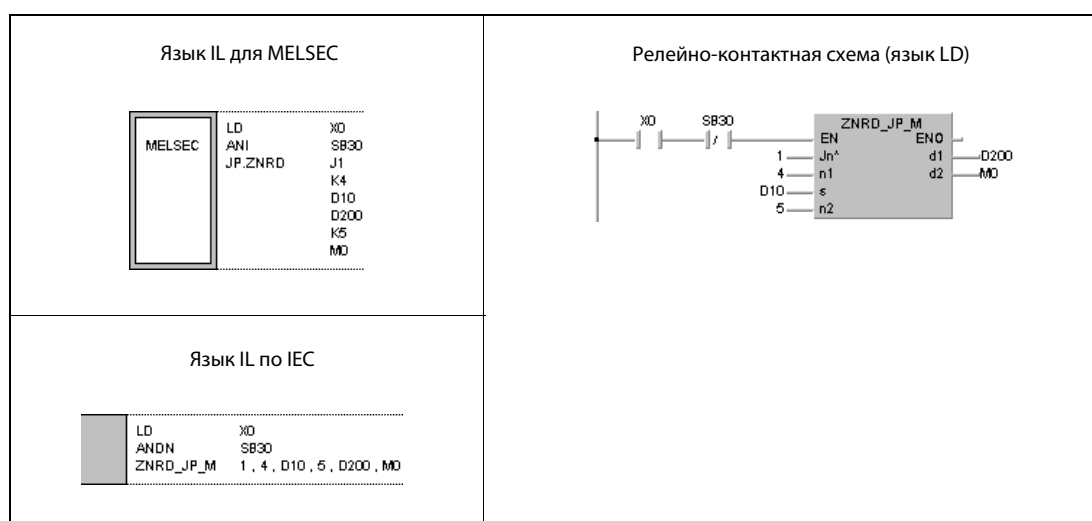
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанная в n2 длина принимаемых данных находится вне области операнда s1, предусмотренной для сохранения (код ошибки 4101).
- Сеть с номером Jn не существует (код ошибки 4102).
- Станция с номером, указанным в n1, не существует (код ошибки 4102).
- Указанная в n2 длина принимаемых данных не находится в диапазоне 1...230 (код ошибки 4100).

**Пример**

JP.ZNRD

Следующая программа при положительном фронте сигнала X0 считывает данные из регистров D10...D14 станции номер 4 и сохраняет считанные данные в регистрах D200...D204 хост-станции. Хост-станция и целевая станция находятся в сети 1.



8.7.2 ZNWR

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	

Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (сист., польз.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
n1	●	●	●	—	—	—	—	●	—	SM0	32
d1	—	● <sup>1</sup>	—	—	—	—	—	—	—		
s	—	●	●	—	—	—	—	—	—		
n2	—	●	●	—	—	—	—	●	—		
d2	●	●	●	—	—	—	—	—	—		

<sup>1</sup> только T, C, D, и W

GX IEC Developer

<p>Язык IL для MELSEC</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p>MELSEC</p> </div> <p>J.ZNWR      Jn n1 d1 s n2 d2</p>	<p>Релейно-конт. схема (язык LD)</p>	<p>Язык IL по IEC</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p>ZNWR_J_M      Jn , n1 , s , n2 , d1 , d2</p> </div>
--	--------------------------------------	--

Переменные

Операнд	Значение	Тип данных
Jn	Номер сети хост-станции ● <sup>1</sup>	BIN, 16 бит
n1	Номер целевой станции	
d1	Первый операнд целевой станции, в который записываются данные.	
s	Первый операнд хост-станции, в котором сохранены записываемые данные.	
n2	Длина передаваемых данных	
d2	Операнд, устанавливаемый после полной обработки команды на один программный цикл.	бит

ПРИМЕЧАНИЕ

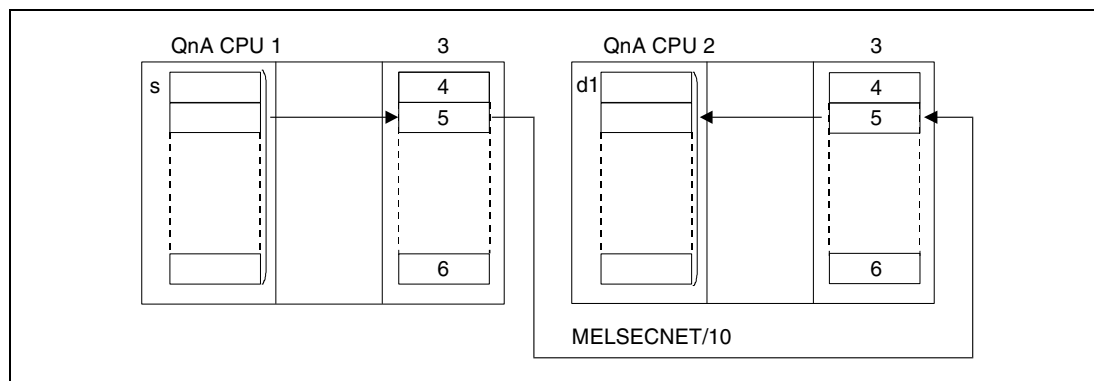
●<sup>1</sup>Если эта команда используется в MELSECNET(I/II/B), следует указать J0.



**Принцип действия****Запись данных в другие станции****ZNWR Команда записи**

Команда ZNWR записывает указанное в n2 количество слов данных, хранящихся в хост-станции в s, в целевую станцию, подключенную к MELSECNET/10. Номер целевой станции указан в n1. Номер сети указан в Jn.

По окончании операции записи в целевой станции устанавливается операнд, указанный в d2.



- 1 хост-станция
- 2 целевая станция
- 3 модуль сетевой коммуникации
- 4 канал 1
- 5 канал 2
- 6 канал 8

Операция записи возможна только в отношении целевой станции, подключенной к той же сети MELSECNET/10, что и хост-станция.

Операция записи с локальной станции возможна только в отношении мастер-станции в сети MELSECNET.

Номера сетей в Jn можно указывать в диапазоне от 1 до 239. Указание номера сети 0 (J0) равносильно указанию MELSECNET в качестве системы сетевой коммуникации.

В системе MELSECNET номер целевой сети (Jn) имеет фиксированную настройку 0 (J0). Номера целевых сетей (Jn) от 1 до 239 используются в MELSECNET/10.

В n1 можно указать номер станции между 1 и 64.

При использовании сетевой системы MELSECNET/B номер станции может находиться в диапазоне от 1 до 31.

Указываемая в n2 длина передаваемых данных (количество слов данных) может находиться в диапазоне от 1 до 230.

С помощью команды ZNWR данные можно записывать как в станции, работающие на процессорах AnU, так и в станции, работающие на процессорах QnA.

Для команды ZNWR обе станции – главная (хост) и целевая – используют канал 2 модуля сетевой коммуникации. Если выполняются несколько команд ZNWR, происходит многократное обращение к каналу 2, хотя канал 2 модуля сетевой коммуникации может одновременно использоваться одной командой только один раз. Чтобы предотвратить одновременное выполнение нескольких команд, следует предусмотреть блокировку на основе сигнала запроса считывания/записи и операнда, устанавливаемого после завершения операции.

Выполнение и конец выполнения команды ZNWR отображаются с помощью флага коммуникационного канала (SB32) и операнда (d2), устанавливаемого при завершении операции, следующим образом:

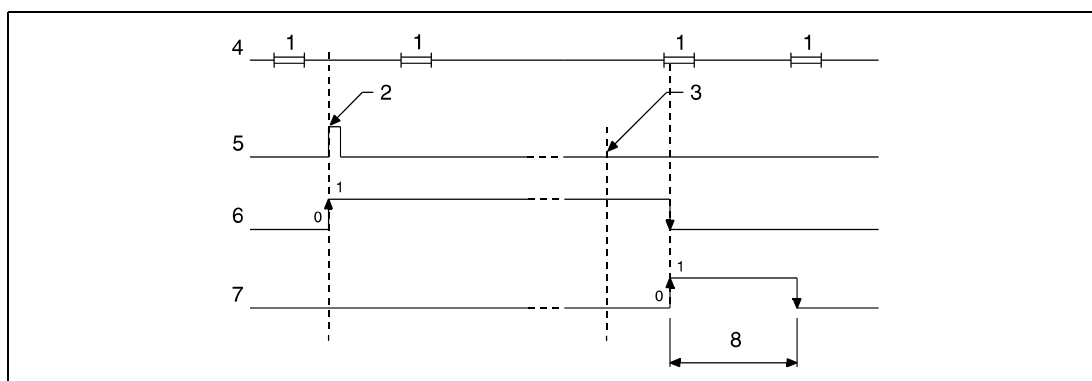
Флаг коммуникационного канала

Этот флаг устанавливается во время выполнения команды ZNWR. Флаг сбрасывается при обработке команды END того цикла, в котором была завершена операция записи.

Операнд хост-станции, показывающий завершение операции записи

Этот операнд устанавливается при обработке команды END того цикла, в котором была завершена операция записи. При обработке следующей команды END этот операнд снова сбрасывается.

На рисунке ниже изображены операции хост-станции во время выполнения команды ZNWR.



- 1 обработка команды END
- 2 выполнение команды ZNWR
- 3 завершение операции
- 4 программа хост-станции
- 5 команда ZNWR
- 6 флаг коммуникационного канала (SB32)
- 7 Операнд хост-станции, устанавливаемый по окончании операции (d2).
- 8 один цикл

Состояние после завершения команды ZNWR (нормальное, не нормальное) отображается с помощью регистра завершения обработки команды ZNWR (SW33) следующим образом:

При нормальном (безошибочном) завершении операции содержимое регистра SW33 равно 0.

При ненормальном (ошибочном) завершении операции в регистре SW33 сохраняется соответствующий код ошибки.

**ПРИМЕЧАНИЕ** *Дополнительная информация о кодах ошибок имеется в руководстве "MELSECNET/10 для сетевых систем QnA".*

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанная в n2 длина передаваемых данных находится вне области операнда s1, предусмотренной для сохранения (код ошибки 4101).
- Сеть с номером Jn не существует (код ошибки 4102).
- Станция с номером, указанным в n1, не существует (код ошибки 4102).
- Указанная в n2 длина передаваемых данных не находится в диапазоне 1...230 (код ошибки 4100).

**Пример**

JP.ZNWR

Следующая программа при положительном фронте сигнала X0 записывает данные из регистров D300...D303 хост-станции в регистры D50...D53 станции 3. Хост-станция и целевая станция находятся в сети 1.



## 8.7.3 LRDP

## Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

## Операнды MELSEC A

	Операнды															Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9011																	
	битовые					словные (16 бит)					Константы		Указатели		Уровень																						
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1							Z	V	K	H (16#)	P	I	N										
n1																	●	●																			
s								●	●	●	●																				11			●			●
d								●	●	●	●																										
n2																	●	●																			

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

## GX IEC Developer



## Переменные

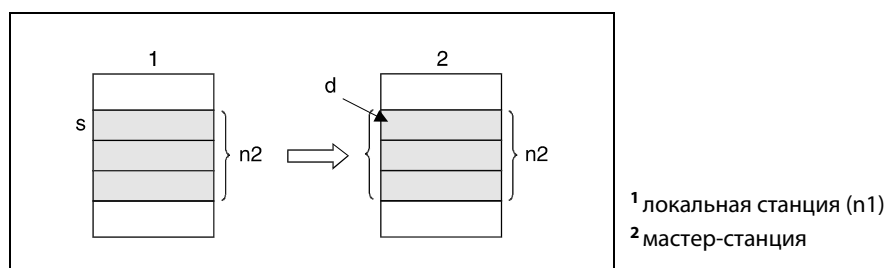
Операнд	Значение	Тип данных
n1	Номер локальной станции	BIN, 16 бит
s	Начальный адрес считываемой области данных локальной станции	
d	Диапазон адресов мастер-станции, в котором сохраняются считанные данные.	
n2	Длина принимаемых данных	

**Принцип действия** **Считывание данных из локальной станции****LRDP Команда считывания**

Команда LRDP считывает данные из локальной станции и сохраняет их в заданном диапазоне адресов мастер-станции. Начальный адрес считываемой области данных задается в  $s$ , количество данных (от 1 до 32) – в  $n2$ , а номер локальной станции – в  $n1$ . Диапазон адресов мастер-станции, в котором должны сохраняться данные, устанавливается в  $d$ .

Во время выполнения команды LRDP в мастер-станции установлен специальный маркер M9200. По окончании выполнения устанавливается M9201. Оба специальных маркера остаются установленными и после обработки команды. Сброс должен осуществляться в основной программе.

Одновременное выполнение двух или более команд LRDP не возможно. Кроме того, команда LRDP не может получить доступ к локальной станции одновременно с командой LWTP.

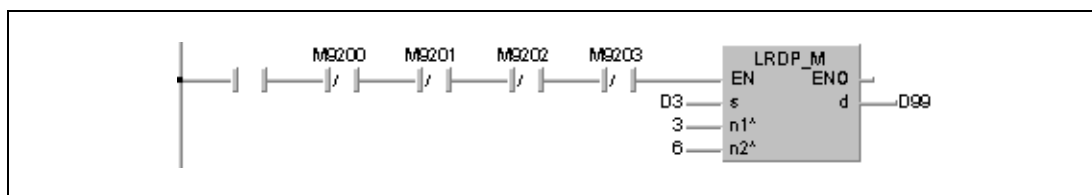


**ПРИМЕЧАНИЕ** В качестве блокировки и входного условия команды LRDP или LWTP следует запрограммировать специальные маркеры M9200, M9201, M9202 и M9203 – так, чтобы было исключено выполнение другой команды LRDP или LWTP.

Результат выполнения команды LRDP отображается значением регистра D9200 в мастер-станции (см. следующую таблицу).

Содержимое регистра	Значение
<b>LRDP D9200</b>	
0	Выполнение завершено без ошибок
2	Ошибка обработки из-за ошибочной адресации: Адреса $s$ и $d$ находятся вне областей операндов, предусмотренных для сохранения. Значение $n1$ не находится в диапазоне 1...64. Значение $n2$ не находится в диапазоне 1...32.
3	Целевая локальная станция находится в состоянии офлайн, поэтому доступ к ней не возможен.
4	По указанному номеру станции нет локальной станции (обработка ошибки).

На рисунке ниже показана блокировка команды LRDP.



#### Источники ошибок

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

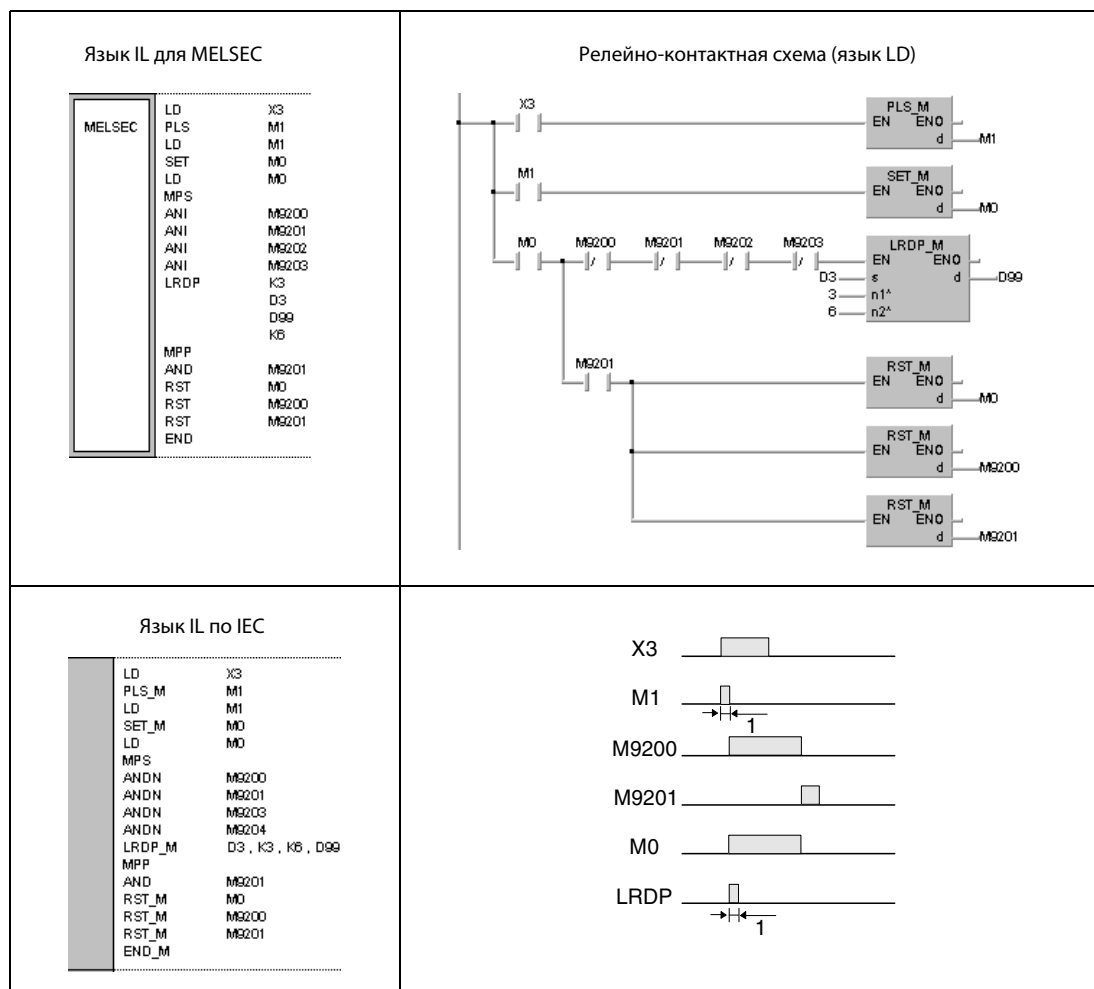
- По указанному в n1 номеру нет локальной станции или указанный номер не находится в диапазоне от 1 до 64.
- Количество указанных в n2 адресов в операндах, указанных в s и d, выходит за пределы предусмотренной для сохранения области операндов.
- Количество указанных в n2 данных не находится в диапазоне от 1 до 32.
- Команда LRDP выполняется в программе локальной станции.

#### ПРИМЕЧАНИЕ

Если центральный процессор не поддерживает коммуникацию или находится в режиме офлайн, команда LRDP не выполняется. Сообщение об ошибке не возникает. Однако маркер M9200 устанавливается.

**Пример** LRDP

Следующая программа считывает данные из регистров D3...D8 локальной станции 3 и сохраняет их в регистрах D99...D104 мастер-станции. После включения входа X3 устанавливается маркер M0 и выполняется команда LRDP. При начале передачи данных устанавливается маркер M9200. По окончании передачи устанавливается M9201. Команда LRDP не выполняется, если уже выполняется другая команда LRDP или LWTP. По окончании передачи (M9201 установлен) в ходе дальнейшего выполнения программы сбрасываются маркеры M0, M9200 и M9201.



<sup>1</sup> однократное выполнение

**ПРИМЕЧАНИЯ** Относящийся к M1 контакт должен работать в качестве импульсного сигнала, так как в противном случае полное выполнение команды LRDP не возможно.

Относящийся к M0 контакт должен быть установлен с помощью команды SET. Если вместо команды SET запрограммирована команда OUT или PLS, при выполнении команды LRDP могут возникнуть ошибки.

Чтобы исключить одновременное выполнение двух команд LRDP, в качестве блокировки следует запрограммировать специальные маркеры M9200 и M9201.

Если в той же программе требуется обращаться к локальной станции с помощью команды LWTP, в качестве блокировки необходимо дополнительно запрограммировать специальные маркеры M9202 и M9203.

### 8.7.4 LWTP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

**Операнды MELSEC A**

	Операнды															Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9011	
	битовые					словные (16 бит)					Константы		Указатели		Уровень						
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V
n1																	●	●			
d							●	●	●	●											
s							●	●	●	●											
n2																	●	●			

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

**GX IEC Developer**



**Переменные**

Операнд	Значение	Тип данных
n1	Номер локальной станции	BIN, 16 бит
d	Диапазон адресов локальной станции, в который записываются данные.	
s	Начальный адрес записываемой области данных мастер-станции.	
n2	Длина передаваемых данных	

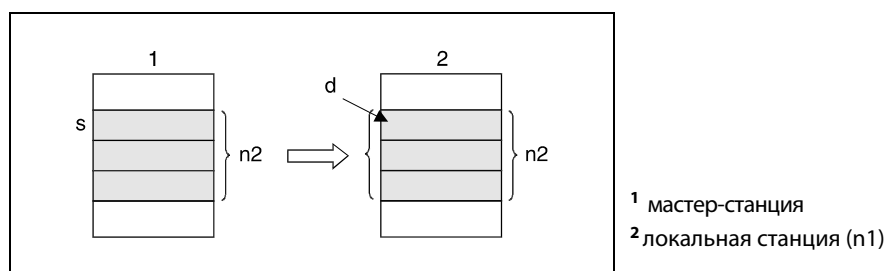


**Принцип действия****Запись данных в локальную станцию****LWTP Команда записи**

Команда LWTP записывает данные мастер-станции в заданную область адресов локальной станции. Начальный адрес передаваемых данных задается в *s*, количество данных (от 1 до 32) – в *n2*, а номер локальной станции – в *n1*. Область адресов локальной станции, в которую требуется передать данные, устанавливается в *d*.

Во время выполнения команды LWTP в мастер-станции установлен специальный маркер M9202. По окончании выполнения устанавливается M9203. Оба специальных маркера остаются установленными и после обработки команды. Сброс должен осуществляться в основной программе.

Одновременное выполнение двух или более команд LWTP не возможно. Кроме того, команда LWTP не может обращаться к локальной станции одновременно с командой LRDP.

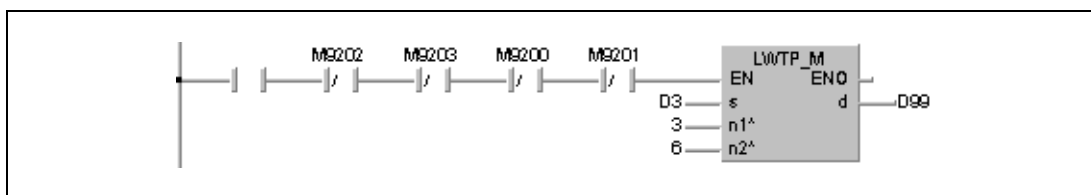
**ПРИМЕЧАНИЕ**

В качестве блокировки и входного условия команды LRDP или LWTP следует запрограммировать специальные маркеры M9200, M9201, M9202 и M9203 – так, чтобы было исключено выполнение другой команды LRDP или LWTP.

Результат выполнения команды LWTP отображается значением регистра D9001 в мастер-станции (см. следующую таблицу).

Значение	Смысл
<b>LWTP D9201</b>	
0	Выполнение завершено без ошибок.
2	Ошибка обработки из-за ошибочной адресации: Адреса <i>s</i> и <i>d</i> находятся вне областей операндов, предусмотренных для сохранения. Значение <i>n1</i> не находится в диапазоне 1...64. Значение <i>n2</i> не находится в диапазоне 1...32.
3	Целевая локальная станция находится в состоянии офлайн, поэтому доступ к ней не возможен.
4	По указанному номеру станции нет локальной станции (обработка ошибки).

На рисунке ниже показана блокировка команда LWTP.



#### Источники ошибок

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

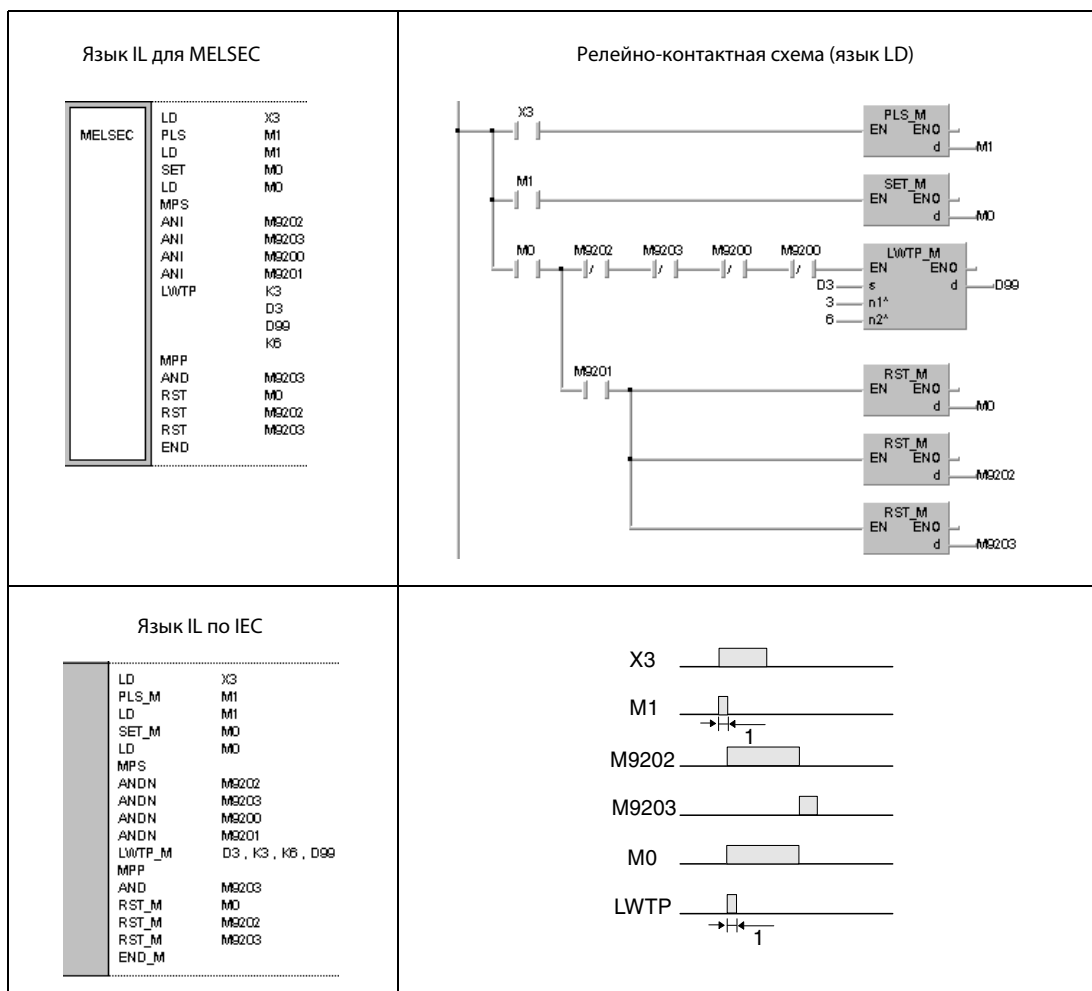
- По указанному в n1 номеру нет локальной станции или указанный номер не находится в диапазоне от 1 до 64.
- Количество указанных в n2 адресов в операндах, указанных в s и d, выходит за пределы предусмотренной для сохранения области операндов.
- Количество указанных в n2 данных не находится в диапазоне от 1 до 32.
- Команда LWTP выполняется в программе локальной станции.

#### ПРИМЕЧАНИЕ

Если центральный процессор не поддерживает коммуникацию или находится в режиме офлайн, команда LWTP не выполняется. Сообщение об ошибке не возникает. Однако M9202 устанавливается.

**Пример** LWTP

Следующая программа передает данные из регистров D99...D104 мастер-станции в регистры D3...D8 локальной станции 3. После включения входа X3 устанавливается маркер M0 и выполняется команда LWTP. При начале передачи данных устанавливается маркер M9202. По окончании передачи устанавливается маркер M9203. Команда LWTP не выполняется, если уже выполняется другая команда LWTP или LRDP. По окончании передачи в ходе дальнейшего выполнения программы сбрасываются маркеры M0, M9202 и M9203.



<sup>1</sup> однократное выполнение

**ПРИМЕЧАНИЯ** Относящийся к M1 контакт должен работать в качестве импульсного сигнала, так как в противном случае полное выполнение команды LWTP не возможно.

Относящийся к M0 контакт должен быть установлен с помощью команды SET. Если вместо команды SET запрограммирована команда OUT или PLS, при выполнении команды LWTP могут возникнуть ошибки.

Чтобы исключить одновременное выполнение двух команд LWTP, следует в качестве блокировки запрограммировать специальные маркеры M9202 и M9203.

Если в той же программе требуется обращаться к локальной станции с помощью команды LRDP, необходимо дополнительно в качестве блокировки запрограммировать специальные маркеры M9200 и M9201.

8.7.5 RFRP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	

Операнды MELSEC A

	Операнды															Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки					
	битовые					словные (16 бит)					Константы		Указатели		Уровень										
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1							Z	V	K	H (16#)	P
n1																	●	●							
n2																	●	●							
d										●													● <sup>1</sup>	●	●
n3																	●	●							

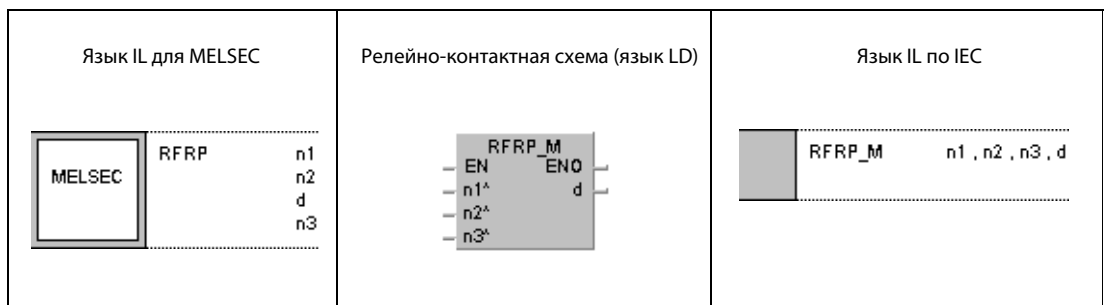
<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

Операнды MELSEC Q

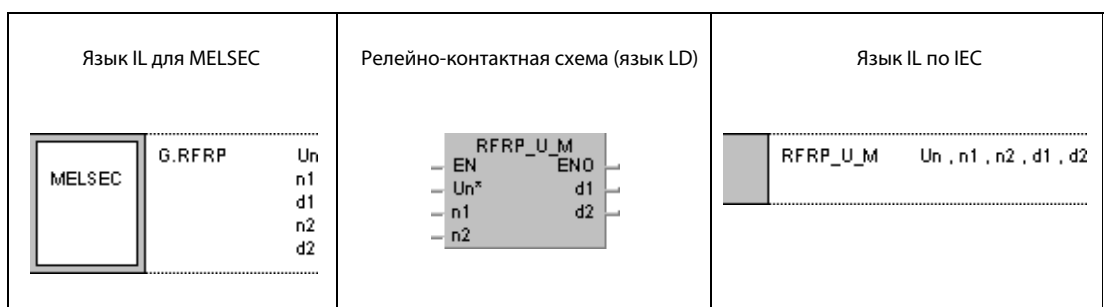
	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (сист., польз.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
n1	●	●	●	—	—	—	—	●	—	SM0	11
d1	—	● <sup>1</sup>	—	—	—	—	—	—	—		
n2	●	●	●	—	—	—	—	●	—		
d2	●	●	●	—	—	—	—	—	—		

<sup>1</sup> только регистры связи

Серия "А" GX IEC Developer



Серия "Q" GX IEC Developer



## Переменные

Операнд		Значение	Тип данных
серия "А"	серия "Q"		
n1	Un	Головной адрес ввода-вывода специального модуля в удаленной станции ввода-вывода ● <sup>1</sup>	BIN, 16 бит
n2	n1	Первый адрес буферной памяти специального модуля, в котором сохранены считываемые данные.	
d	d1	Первый адрес регистров связи хост-станции, в которых сохраняются считанные данные.	адрес
n3	n2	Длина принимаемых данных	BIN, 16 бит
	d2	Операнд, устанавливаемый после полной обработки команды на один программный цикл.	бит

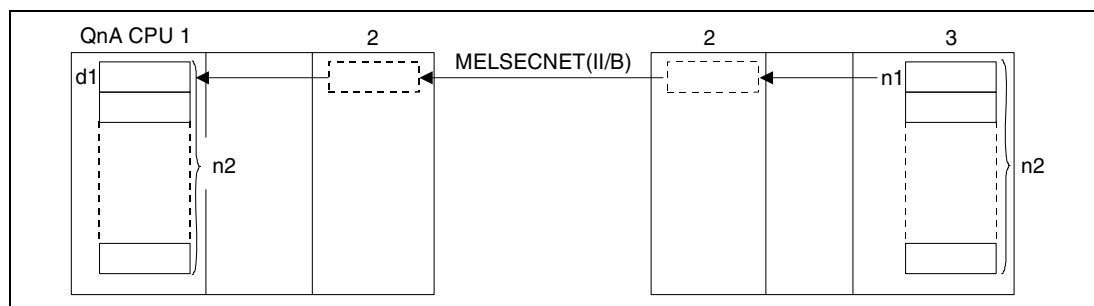
**ПРИМЕЧАНИЕ** ●<sup>1</sup> Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции должен находиться в диапазоне от 0 до FEn. Следует иметь в виду, что в Un компилятор ожидает шестнадцатеричного указания адреса. Десятичный адрес автоматически преобразуется в шестнадцатеричное число.

**Принцип действия****Считывание данных из удаленной станции****RFRP Команда считывания**

Команда RFRP считывает данные из буферной памяти специального модуля удаленной станции в сети MELSECNET. Количество считываемых слов данных указано в n2 (серия "А" = n3). Диапазон адресов буферной памяти указан, начиная с n1 (серия "А" = n2).

Адрес ввода-вывода подключенного специального модуля указан в Un (серия "А" = n1). Считанные данные сохраняются в регистрах связи мастер-станции, начиная с d1 (серия "А" = d).

По окончании операции считывания в удаленной станции ввода-вывода устанавливается операнд, указанный в d2 (только серия QnA).



1 главная станция (хост/мастер)

2 модуль коммуникации

3 специальный модуль (целевая станция/удаленная станция ввода-вывода)

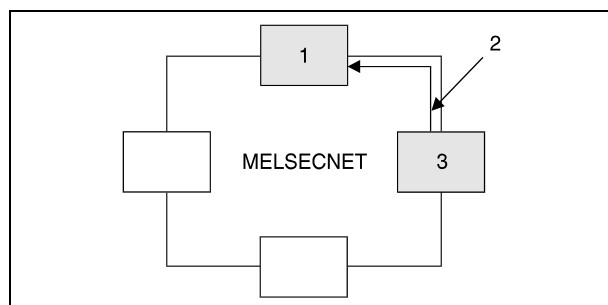
**ПРИМЕЧАНИЕ**

Даже если выполняется только одна операция считывания, диапазон адресов регистров связи в d1 (серия "А" = d) должен находиться в пределах, установленных при параметрировании сети MELSECNET в удаленной станции и мастер-станции.

В следующих описаниях адресов ввода-вывода специального модуля для примера используются адреса серии QnA. Эти описания применимы и к серии "А", однако вместо значения "n" следует подставлять "n1" (например, серия QnA = Y(n + E) ⇒ серия "А" = Y(n1 + E)).

Во время выполнения команды RFRP установлен выход Y(n + E). Вход X(n + 1E) устанавливается сразу по окончании выполнения команды. Y(n + E) остается установленным и после окончания выполнения, и поэтому его необходимо отдельно сбрасывать в программе. Адресация происходит автоматически и ее нельзя изменять.

Операции считывания из удаленных станций ввода-вывода можно выполнять с мастер-станции, подключенной к сети MELSECNET.



1 станция, выполняющая команду RFRP (мастер)

2 операция считывания данных из специального модуля

3 удаленная станция ввода-вывода

Если выполнение команды RFRP не возможно из-за неполадки в специальном модуле удаленной станции, устанавливается X(n + 1D). В этом случае необходимо проверить соответствующий модуль. X(n + 1D) сбрасывается, как только устанавливается Y(n + D).

В случае 4-значного отображения указанные в Un головные адреса ввода-вывода специальных модулей указываются 3 старшими разрядами. Например, для указания адресов X/Y0200 необходимо указать значение 20 (только серия "Q").

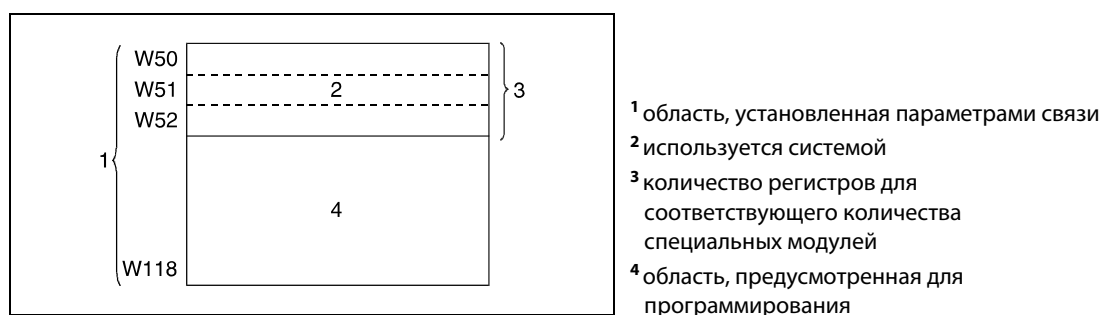
**ПРИМЕЧАНИЕ** *Дополнительную информацию о возможной области адресов буферной памяти в специальном модуле, которую требуется указывать в n1 (серия "A" = n2), можно найти в руководстве по соответствующему специальному модулю.*

Длина принимаемых данных (количество слов данных), указываемая в n2 (серия "A" = n3), может находиться в диапазоне между 1 и 16.

Диапазон адресов регистров связи в d1 (серия "A" = d) должен находиться в пределах области, выделенной для связи в параметрах удаленной станции и мастер-станции.

Необходимо четко различать диапазон регистров связи Wxxx главной станции и диапазон регистров удаленной станции. Количество адресов регистров связи, используемых операционной системой, соответствует количеству специальных модулей, находящихся в удаленных станциях сети. Чтобы определить полезную область, которую можно использовать для сохранения данных, следует из области параметров вычесть адреса регистров связи, используемые операционной системой.

Распределение области регистров связи пояснено изображенным ниже примером. Область между мастер-станцией и удаленной станцией установлена в параметрах с W050 по W118 (серия "A" = W09F). В этой области находятся 2 специальных модуля, поэтому первые два регистра связи W50 и W51 (2 адреса) используются операционной системой центрального процессора. Соответственно, полезный диапазон для сохранения данных находится между W52 и W118 (серия "A" = W09F).



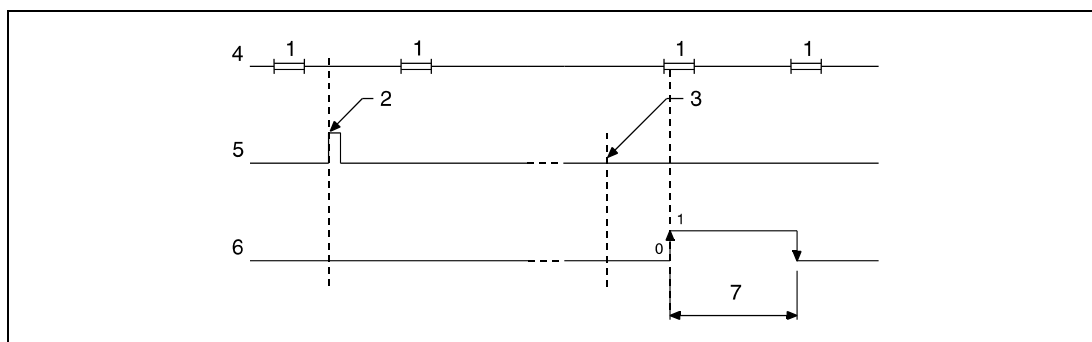
Команды RFRP и RTOP не могут выполняться одновременно из нескольких источников в одном и том же специальном модуле. При одновременном выполнении команд из двух или более источников выполнение других команд RFRP и RTOP предотвращается благодаря режиму квитирования (handshake) обеих активных станций.

В качестве блокировки команды RFRP входы и выходы X(n + 1E) и Y(n + E) следует запрограммировать так, чтобы было исключено одновременное выполнение другой команды RFRP или RTOP.

Операнд хост-станции, показывающий завершение операции считывания (d2), устанавливается при обработке команды END того цикла, в котором завершена операция считывания. При обработке следующей команды END этот операнд снова сбрасывается (только серия "Q").

Контроллеры MELSEC серии "А" для передачи данных по сети MELSECNET имеют ряд специальных регистров, регистрирующих различные состояния коммуникации. Так, например, состояние удаленных станций ввода-вывода регистрируется с помощью специальных регистров с D9228 по D9231. Обращения к параметрам оцениваются с помощью специальных маркеров с M9224 по M9227 (только серия "А").

На рисунке ниже изображены операции хост-станции во время выполнения команды RFRP.



- 1 обработка команды END
- 2 выполнение команды RFRP
- 3 завершение операции
- 4 программа хост-станции
- 5 команда RFRP
- 6 операнд хост-станции, устанавливаемый по окончании операции (d2) (только серия "Q")
- 7 один цикл

#### Источники ошибок

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

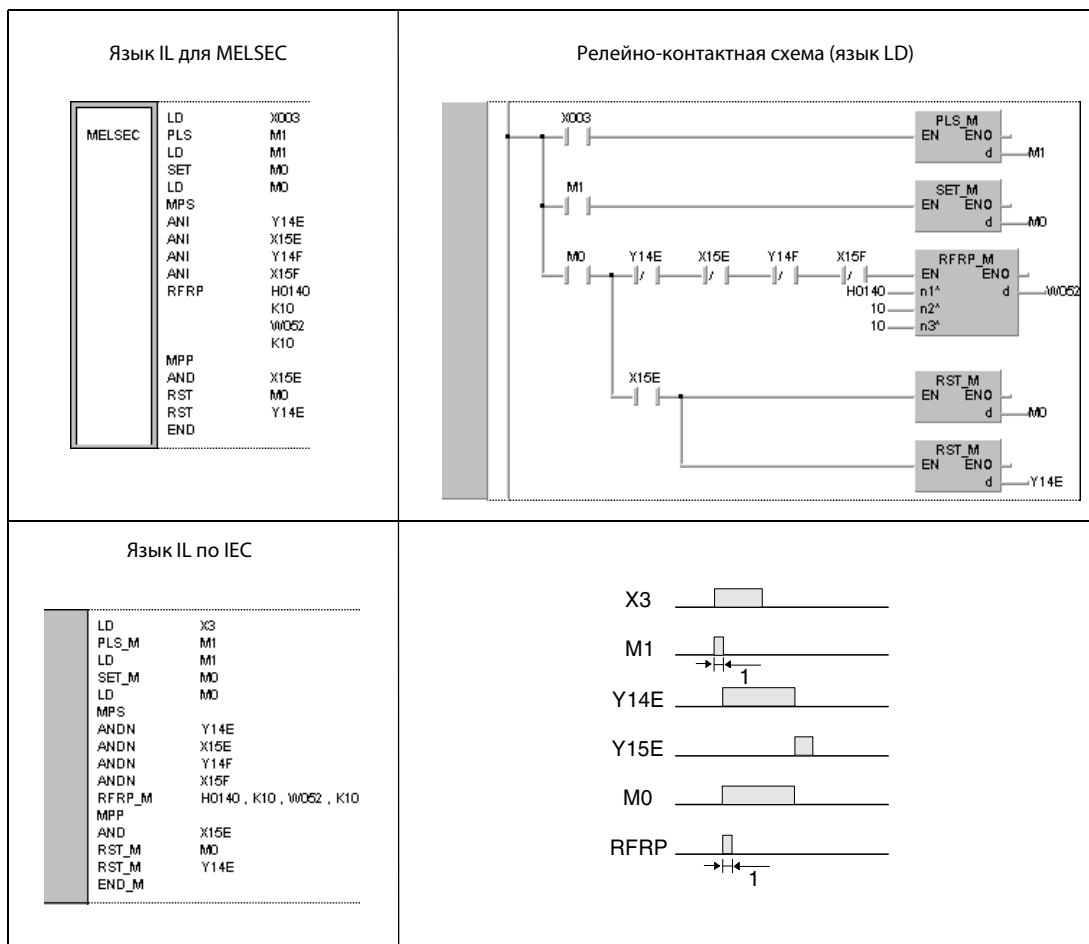
- Адрес ввода-вывода, указанный в Un (серия "А" = n1), не является адресом удаленной станции ввода-вывода (серия "Q" = код ошибки 4102).
- Указанный в n1 (серия "А" = n2) адрес ввода-вывода не является головным адресом ввода-вывода специального модуля (код ошибки 4102).
- Указанное в n2 (серия "А" = n3) количество адресов находится вне возможного диапазона адресов, начиная с d1 (серия "А" = d, W0...W3FF) (серия "Q" = код ошибки 4101).
- Сеть, указанная в Un (серия "А" = n1), не существует (код ошибки 2413).
- Значение, указанное в n2 (серия "А" = n3), не находится в диапазоне от 1 до 16 (код ошибки 4100).



**Пример** RFRP (серия "А")

Следующая программа считывает данные из 10 адресов специального модуля (например, А68АD) подряд, начиная с адреса 10. Модуль находится во второй удаленной станции. Диапазон адресов составляет от 140 до 15F. Считанные данные сохраняются в регистрах связи с W52 по W61 мастер-станции.

После включения входа X3 устанавливается маркер M0 и выполняется команда RFRP. При начале передачи данных устанавливается  $Y(n + E) = Y14E$ . По окончании передачи устанавливается  $X(n + 1E) = X15E$ . Команда RFRP не выполняется, если уже выполняется другая команда RFRP или RTOP. По окончании передачи, в ходе дальнейшего выполнения программы маркеры M0 и Y14E сбрасываются.



<sup>1</sup> однократное выполнение

**ПРИМЕЧАНИЯ** Относящийся к M1 контакт должен работать в качестве импульсного сигнала, так как в противном случае полное выполнение команды RFRP не возможно.

Относящийся к M0 контакт должен быть установлен с помощью команды SET. Если вместо команды SET запрограммирована команда OUT или PLS, то при выполнении команды RFRP могут возникнуть ошибки.

Чтобы исключить одновременное выполнение двух команд RFRP, необходимо в качестве блокировки запрограммировать выход Y14E и вход X15E.

Если в той же программе к этой станции требуется обращаться с помощью команды RTOP, то дополнительно в качестве блокировки необходимо запрограммировать выход Y14F и вход X15F.

8.7.6 RTOP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●	●	

Операнды MELSEC A

	Операнды																Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки	
	битовые						словные (16 бит)						Константы		Указатели							Уровень
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						
n1																	●	●				
n2																	●	●				
s										●												
n3																	●	●				

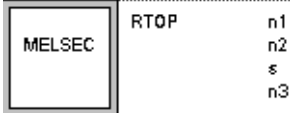
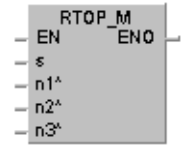

<sup>1</sup> Количество шагов при использовании процессоров AnA, AnAS и AnU указано в разд. 3.10.2 „В случае процессоров AnA, AnAS и AnU“ этого руководства.

Операнды MELSEC Q

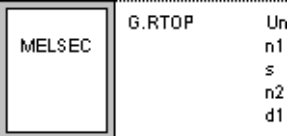
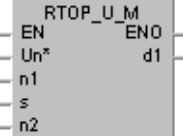
	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (сист., польз.)		Регистры файлов	MELSECNET/10 непос. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
n1	●	●	●	—	—	—	—	●	—	SM0	11
s	—	● <sup>1</sup>	—	—	—	—	—	—	—		
n2	●	●	●	—	—	—	—	—	—		
d	●	●	●	—	—	—	—	●	—		

<sup>1</sup> только регистры связи

Серия "А" GX IEC Developer

<p>Язык IL для MELSEC</p> 	<p>Релейно-контактная схема (язык LD)</p> 	<p>Язык IL по IEC</p> 
---	---	---

Серия "Q" GX IEC Developer

<p>Язык IL для MELSEC</p> 	<p>Релейно-контактная схема (язык LD)</p> 	<p>Язык IL по IEC</p> 
---	---	---

## Переменные

Операнд		Значение	Тип данных
серия "А"	серия "Q"		
n1	Un	Головной адрес ввода-вывода специального модуля в удаленной станции ввода-вывода ● <sup>1</sup>	BIN, 16 бит
n2	n1	Первый адрес буферной памяти специального модуля, в который записываются данные.	
s	s	Первый адрес регистров связи хост-станции, в которых сохранены записываемые данные.	адрес
n3	n2	Длина передаваемых данных	BIN, 16 бит
	d	Операнд, устанавливаемый после полной обработки команды на один программный цикл.	бит

**ПРИМЕЧАНИЕ** ●<sup>1</sup> Головной адрес ввода-вывода модуля сетевой коммуникации хост-станции должен находиться в диапазоне от 0 до FEn. Следует иметь в виду, что в Un компилятор ожидает шестнадцатеричного указания адреса. Десятичный адрес автоматически преобразуется в шестнадцатеричное число.

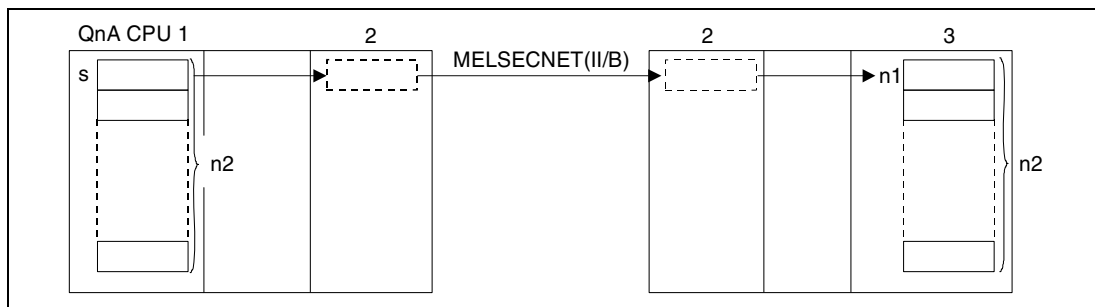
**Принцип действия**

**Запись данных в удаленную станцию**

**RTOP Команда записи**

Команда RTOP записывает данные в буферную память специального модуля удаленной станции, подключенной к сети MELSECNET. Количество записываемых слов данных указано в n2 (серия "А" = n3). Диапазон адресов буферной памяти указан начиная с n1 (серия "А" = n2). Адрес ввода-вывода подключенного специального модуля указан в Un (серия "А" = n1). Записываемые данные хранятся, начиная с регистра связи мастер-станции, указанного в s.

По окончании операции записи в удаленной станции ввода-вывода устанавливается операнд, указанный в d (только серия "Q").



- 1 главная станция (хост/мастер)
- 2 модуль коммуникации
- 3 специальный модуль (целевая станция/удаленная станция ввода-вывода)

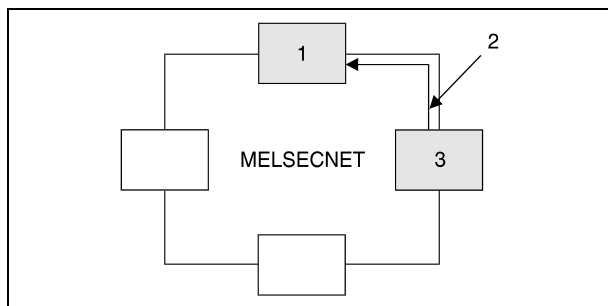
**ПРИМЕЧАНИЕ**

Даже если выполняется только одна операция записи, диапазон адресов регистров связи в s должен находиться в пределах области, выделенной при параметрировании сети MELSECNET в удаленной станции и мастер-станции.

В следующих описаниях адресов ввода-вывода специального модуля для примера используются адреса серии QnA. Эти описания применимы и к серии "А", однако вместо значения "n" следует подставлять "n1" (например, серия QnA = Y(n + F) ⇒ серия "А" = Y(n1 + F)).

Во время выполнения команды RTOP установлен выход Y(n + F). Вход X(n + 1F) устанавливается по окончании выполнения команды. Y(n + F) остается установленным и после окончания выполнения, и поэтому его необходимо сбрасывать в программе отдельно. Адресация происходит автоматически и ее нельзя изменять.

Операции записи в удаленные станции ввода-вывода можно выполнять с мастер-станции, подключенной к сети MELSECNET.



- 1 станция, выполняющая команду RTOP (мастер)
- 2 операция записи данных в специальный модуль
- 3 удаленная станция ввода-вывода

Если выполнение команды RTOP не возможно из-за неполадки в специальном модуле удаленной станции, устанавливается X(n + 1D). В этом случае необходимо проверить соответствующий модуль. X(n + 1D) сбрасывается, как только устанавливается Y(n + D).

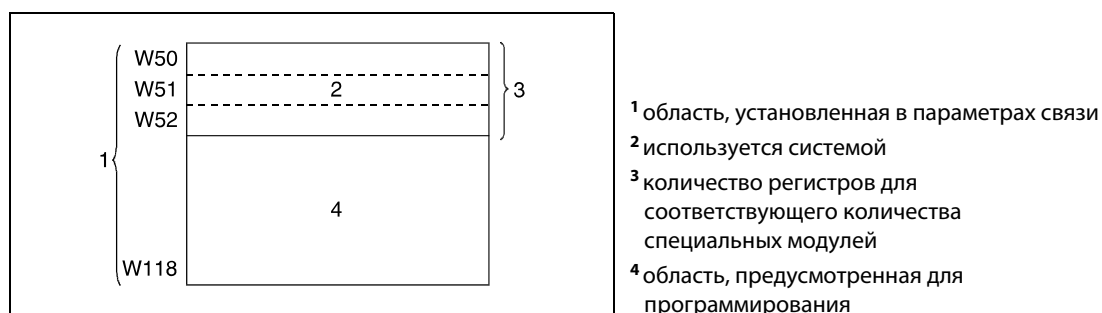
В случае 4-значного отображения указанные в Un головные адреса ввода-вывода специальных модулей указываются 3 старшими разрядами. Например, для указания адресов X/Y0200 необходимо указать значение 20 (только серия "Q").

**ПРИМЕЧАНИЕ** *Дополнительную информацию о возможном диапазоне адресов буферной памяти в специальных модулях, который требуется указывать в n1 (серия "А" = n2), можно найти в руководстве по соответствующему специальному модулю.*

Указываемая в n2 (серия "А" = n3) длина передаваемых данных (количество слов данных) может находиться в диапазоне от 1 до 16.

Необходимо четко различать диапазон регистров связи Wxxx главной станции и диапазон регистров удаленной станции. Количество адресов регистров связи, используемых операционной системой, соответствует количеству специальных модулей, находящихся в удаленных станциях сети. Чтобы определить полезную область, которую можно использовать для сохранения данных, следует из области параметров вычесть адреса регистров связи, используемые операционной системой.

Распределение области регистров связи пояснено изображенным ниже примером. Область между мастер-станцией и удаленной станцией установлена в параметрах с W050 по W118 (серия "А" = W09F). В этой области находятся 2 специальных модуля, поэтому первые два регистра связи W50 и W51 (2 адреса) используются операционной системой центрального процессора. Соответственно, полезный диапазон для сохранения данных находится между W52 и W118 (серия "А" = W09F).



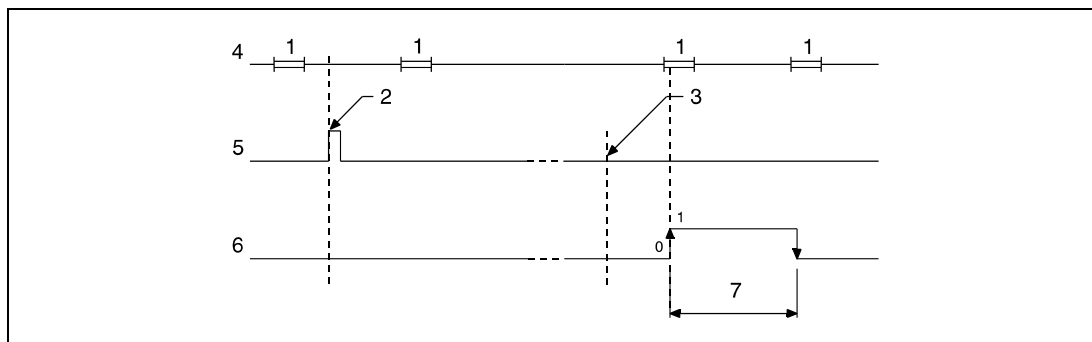
Команды RFRP и RTOP не могут выполняться одновременно из нескольких источников в одном и том же специальном модуле. При одновременном выполнении команд из двух или более источников выполнение других команд RFRP и RTOP предотвращается благодаря режиму квитирования (handshake) обеих активных станций.

В качестве блокировки команды RTOP вход X(n + 1F) и выход Y(n + F) следует запрограммировать так, чтобы было исключено одновременное выполнение другой команды RTOP или RFRP.

Операнд хост-станции, показывающий завершение операции записи (d), устанавливается при обработке команды END того цикла, в котором завершена операция считывания. При обработке следующей команды END этот операнд снова сбрасывается (только серия "Q").

Контроллеры MELSEC серии "А" для передачи данных по сети MELSECNET имеют ряд специальных регистров, регистрирующих различные состояния коммуникации. Так, например, состояние удаленных станций ввода-вывода регистрируется с помощью специальных регистров с D9228 по D9231. Обращения к параметрам оцениваются с помощью специальных маркеров с M9224 по M9227 (только серия "А").

На рисунке ниже изображены операции хост-станции во время выполнения команды RTOP.



- 1 обработка команды END
- 2 выполнение команды RTOP
- 3 завершение операции
- 4 программа хост-станции
- 5 команда RTOP
- 6 операнд хост-станции, устанавливаемый по окончании операции (d)(только серия "Q")
- 7 один цикл

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

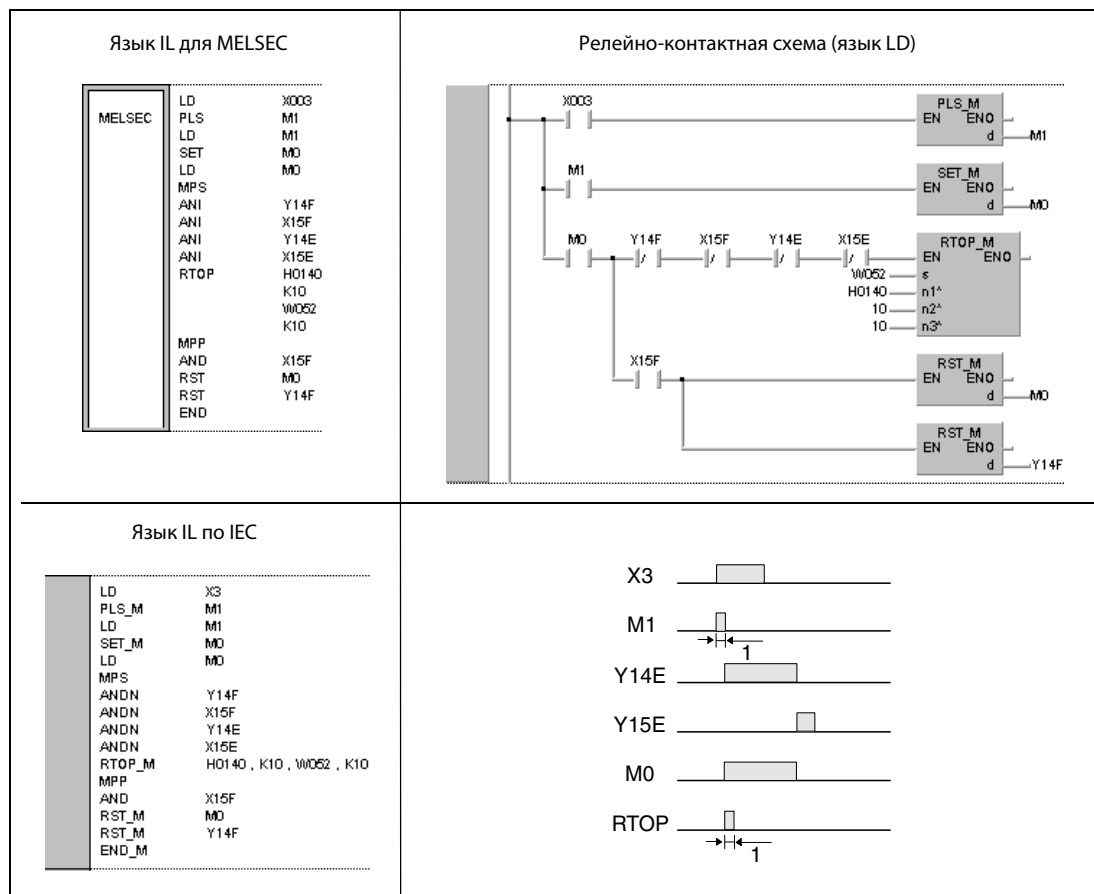
- Адрес ввода-вывода, указанный в Un (серия "А" = n1), не является адресом удаленной станции ввода-вывода (серия "Q" = код ошибки 4102).
- Указанный в n1 (серия "А" = n2) адрес ввода-вывода не является головным адресом ввода-вывода специального модуля (серия "Q" = код ошибки 4102).
- Указанное в n2 (серия "А" = n3) количество адресов находится вне возможного диапазона адресов, начиная с s (серия "А" = W0...W3FF) (серия "Q" = код ошибки 4101).
- Сеть, указанная в Un (серия "А" = n1), не существует (код ошибки 2413).
- Значение, указанное в n2 (серия "А" = n3), не находится в диапазоне от 1 до 16 (код ошибки 4100).

**Пример**

RTOP (серия "A")

Следующая программа записывает данные из регистров связи W52...W61 мастер-станции в 10 следующих друг за другом адресов специального модуля (например, A68AD) во второй удаленной станции. Диапазон адресов составляет от 140 до 15F. Область адресов, в которую требуется передать данные, начинается с адреса 10.

После включения входа X3 устанавливается маркер M0 и выполняется команда RTOP. При начале передачи данных устанавливается  $Y(n1 + F) = Y14F$ . По окончании передачи устанавливается  $X(n1 + 1F) = X15F$ . Команда RTOP не выполняется, если уже выполняется другая команда RTOP или RFRP. По окончании передачи, в ходе дальнейшего выполнения программы сбрасываются маркеры M0 и Y14F.



<sup>1</sup> однократное выполнение

**ПРИМЕЧАНИЯ**

Относящийся к M1 контакт должен работать в качестве импульсного сигнала, так как в противном случае полное выполнение команды RTOP не возможно.

Относящийся к M0 контакт должен быть установлен с помощью команды SET. Если вместо команды SET запрограммирована команда OUT или PLS, при выполнении команды RTOP могут возникнуть ошибки.

Чтобы исключить одновременное выполнение двух команд RTOP, необходимо в качестве блокировки запрограммировать выход Y14F и вход X15F.

Если в той же программе к этой станции требуется обращаться с помощью команды RFRP, то дополнительно в качестве блокировки следует запрограммировать выход Y14E и вход X15E.

## 8.8 Считывание и запись информации маршрутизации

Эти команды позволяют считывать и записывать информацию маршрутизации. Параметры маршрутизации включают в себя номер сети, номер релейной станции и номер станции маршрутизации.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе IEC
Считывание информации маршрутизации	Z.RTREAD	RTREAD_M
	ZP.RTREAD	RTREADP_M
Запись информации маршрутизации	Z.RTWRITE	RTWRITE_M
	ZP.RTWRITE	RTWRITEP_M



### 8.8.1 RTREAD

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

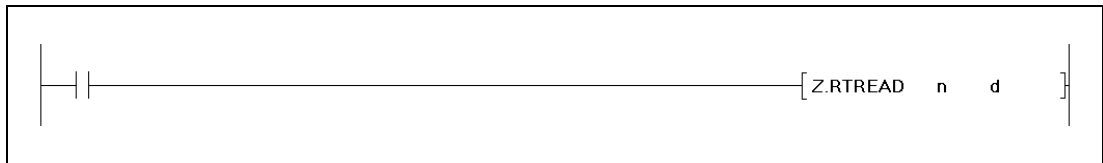
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (сист., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□\G□□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
n	—	●	●	—	—	—	—	●	—	SM0	7
d	—	●	●	—	—	—	—	—	—		

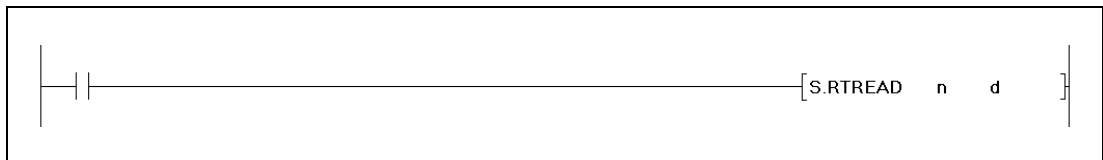
**GX IEC Developer**



**GX Developer (процессоры QnA)**



**GX Developer (System Q)**



**Переменные**

Операнд	Значение	Тип данных	
		MELSEC	IEC
n	Целевая сеть передачи (от 1 до 239)	BIN, 16 бит	ANY16
d	Первый адрес операнда, в котором сохраняется считанная информация маршрутизации.	адрес	массив [1..3] данных типа ANY16

**Принцип действия**

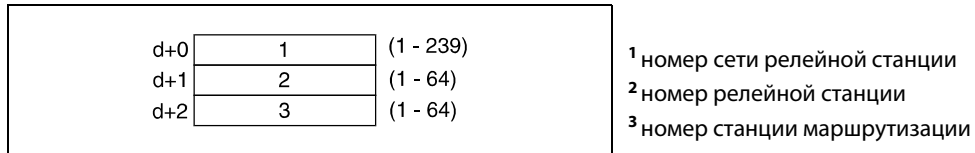
**Считывание информации маршрутизации**

**RTREAD Команда считывания**

Команда RTREAD считывает информацию маршрутизации указанной в n целевой коммуникационной сети. Информация маршрутизации хранится в параметрах маршрутизации. Считанная информация маршрутизации сохраняется начиная с операнда d + 0 (Array\_d[1]).

Если никакие данные для передачи не указаны, то в операнды от d (от Array\_d[1] до Array\_d[3]) записывается значение 0.

Содержимое операндов, начиная с d + 0 (Array\_d[1]), показано на следующем изображении.



**Источники ошибок**

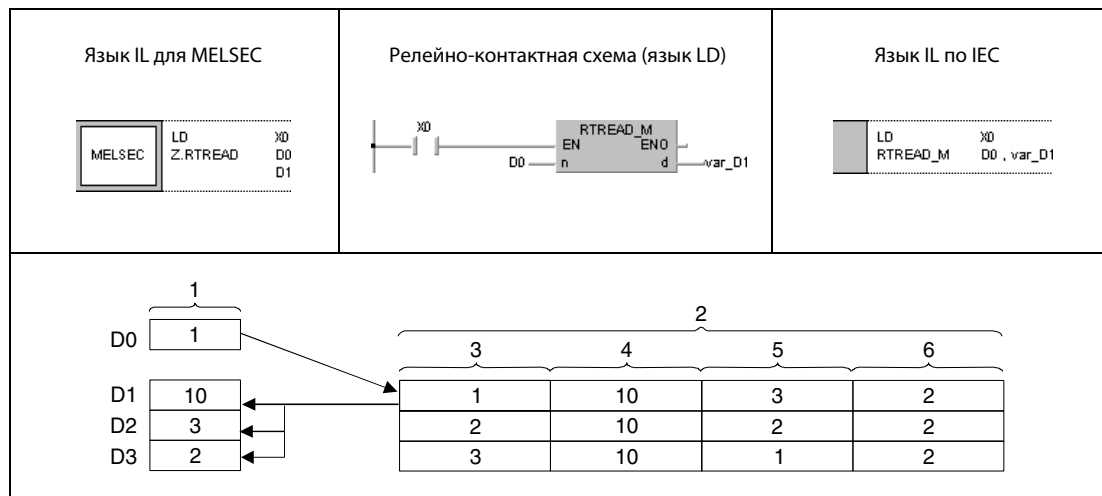
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в n значение данных не находится в диапазоне от 1 до 239 (код ошибки 4100).

**Пример**

**Z.RTREAD**

Следующая программа при включенном входе X0 считывает информацию маршрутизации сети (1), указанной в D0, и сохраняет считанные данные в регистрах с D1 по D3 (от var\_D1[1] до var\_D1[3]).



- 1 операция
- 2 содержимое настроек параметров маршрутизации
- 3 номер целевой сети передачи
- 4 номер сети релейной станции
- 5 номер релейной станции
- 6 номер станции маршрутизации

**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

### 8.8.2 RTWRITE

**Процессор**

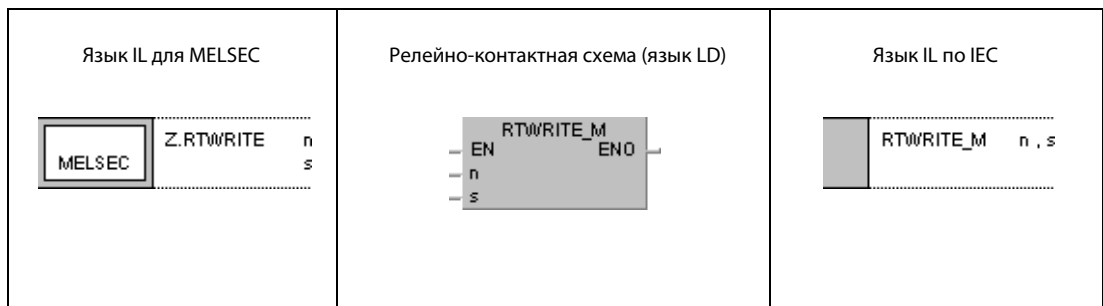
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

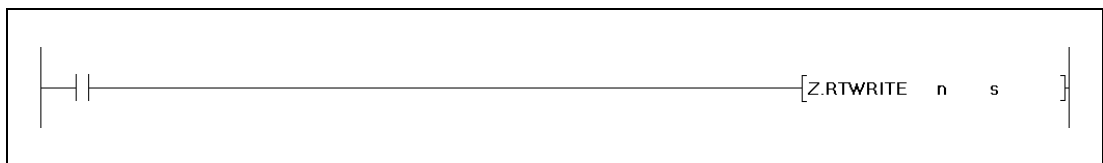
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (сист., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
n	—	●	●	—	—	—	—	●	—	SM0	8
s	—	●	●	—	—	—	—	—	—		

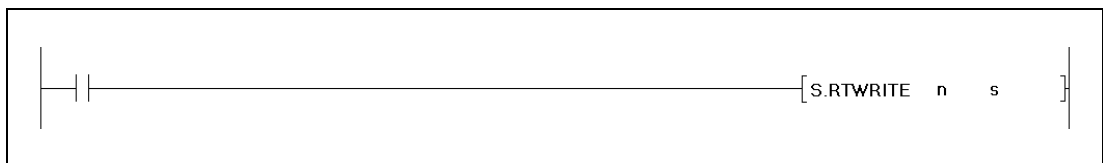
**GX IEC Developer**



**GX Developer (процессоры QnA)**



**GX Developer (System Q)**



**Переменные**

Операнд	Значение	Тип данных	
		MELSEC	IEC
n	Целевая сеть передачи (от 1 до 239)	BIN, 16 бит	ANY16
s	Первый адрес операнда, в котором сохраняется записываемая информация маршрутизации.	адрес	массив [1..3] данных типа ANY16

**Принцип действия**

**Запись информации маршрутизации**

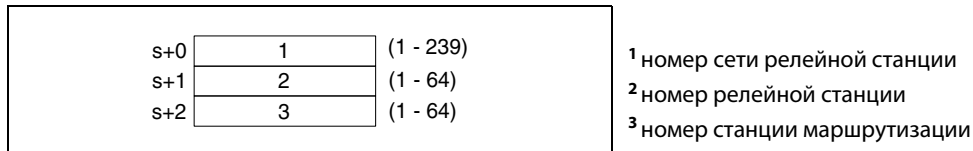
**RTWRITE Команда записи**

Команда RTWRITE записывает информацию маршрутизации для целевой сети, указанной в n. Информация маршрутизации хранится в параметрах маршрутизации, начиная с s + 0 (Array\_s[1]).

Если данные для целевой сети в параметрах маршрутизации установлены, они используются для обновления данных, сохраненных начиная с s + 0 (Array\_s[1]).

Если данные, сохраненные начиная с s (Array\_s[1]), имеют значение 0, то параметры маршрутизации указанной в n целевой сети стираются.

Содержимое данных, указываемых начиная с s + 0 (Array\_s[1]), пояснено на следующем рисунке.



**Источники ошибок**

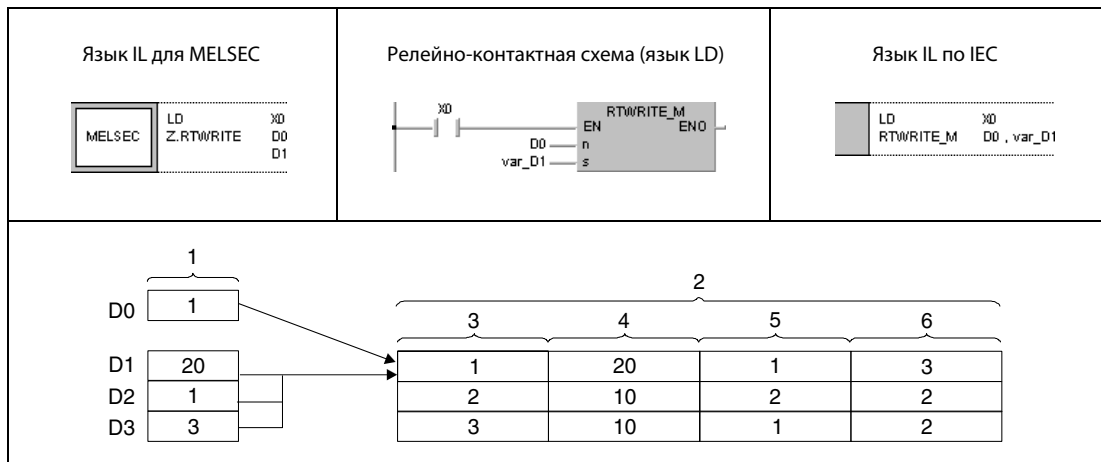
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в n значение данных не находится в диапазоне от 1 до 239 (код ошибки 4100).
- Указанные в s данные находятся вне допустимых диапазонов (код ошибки 4100).

**Пример**

**Z.RTWRITE**

Следующая программа при включенном входе X0 записывает информацию маршрутизации, сохраненную в регистрах с D1 по D3 (от var\_D1[1] до var\_D1[3]), в качестве параметров маршрутизации для сети (1), указанной в D0.



**ПРИМЕЧАНИЕ**

Без объявления переменных в заголовке POU этот пример программы не работоспособен. Компилятор и проверяющая программа могут выдать сообщения об ошибках. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

## 9 Команды для процессоров "System Q"

Описываемые в этом разделе команды применимы только для центральных процессоров "System Q".

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC
Считывание информации модуля	UNIRD	UNIRD_M
	UNIRDP	UNIRDP_M
Распознавание и устранение ошибок	TRACE	TRACE_M
	TRACER	TRACER_M
Перенос данных в файлы и из файлов	FWRITE	FWRITE_M
	FREAD	FREAD_M
Команды для операций с программой	PLOADP	PLOADP_M
	PUNLOADP	PUNLOADP_M
	PSWAPP	PSWAPP_M
Команды передачи	RBMV	RBMV_M
	RBMVOP	RBMVOP_M

При применении процессоров Q02, Q02H, Q06H, Q12H, Q12PH, Q25H и Q25PH, начиная с версии "B" программного обеспечения, можно использовать следующие команды для многопроцессорного режима:

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по IEC
Запись данных в общую область памяти для многопроцессорного режима	S.TO	TO_S_M
	S.TOP	TO_SP_M
Считывание данных из общей области памяти другого процессора в многопроцессорной системе	FROM	FROM_M
	FROMP	FROMP_M

## 9.1 Считывание информации модуля

### 9.1.1 UNIRD, UNIRDP

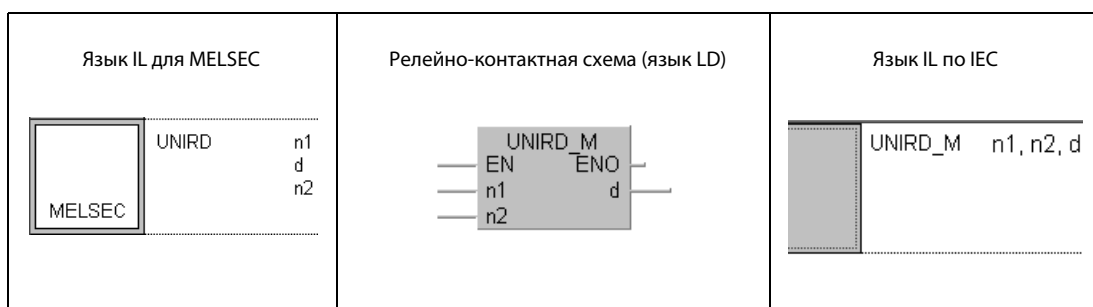
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

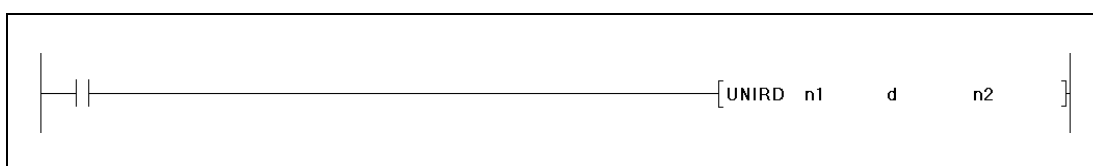
Операнды MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□VG□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
n1	●	●	●	—	—	—	—	●	—	SM0	4
d	—	●	●	—	—	—	—	—	—		
n2	●	●	●	—	—	—	—	●	—		

GX IEC Developer



GX Developer



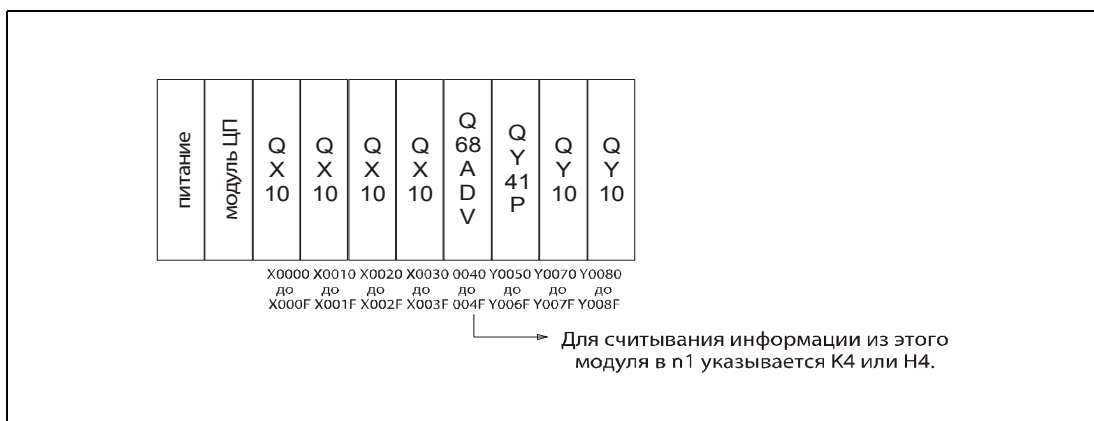
Переменные

Операнд	Значение	Тип данных
n1	Результат деления начального адреса модуля, из которого требуется считать информацию, на 16 (от 0 до FF <sub>h</sub> ).	BIN, 16 бит
d	Первый операнд целевой области, в котором сохраняются данные.	адрес
n2	Количество считываемой информации модуля (от 0 до 256)	BIN, 16 бит

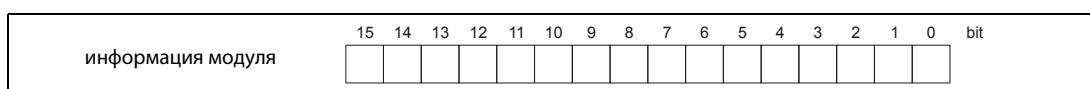
**Принцип действия**      **Считывание информации модуля**  
**UNIRD    Команда считывания**

Информация модуля считывается начиная с адреса ввода-вывода, указанного в n1, и записывается, начиная с адреса, указанного в d. Длина данных указывается в n2. Чтобы получить значение для n1, следует разделить начальный адрес модуля на 16.  
 С помощью команды UNIRD можно определять состояние действительно установленных модулей, независимо от присвоения модулей в параметрах ввода-вывода.

**ПРИМЕЧАНИЕ**      *Значение для n1 образуется из трех старших разрядов начального адреса слота, из которого требуется считать информацию модуля. Весь адрес выражен четырьмя разрядами шестнадцатеричного числа.*



В следующей таблице разъяснено значение отдельных битов информации модуля.



Бит	Значение	Описание
0	Количество входов или выходов	000: 16      001: 32      010: 48      011: 64 100: 128    101: 256    110: 512    111: 1024
1		
2		
3	Тип модуля	000: входной модуль      001: выходной модуль 010: модуль ввода-вывода    011: специальный модуль
4		
5		
6	Состояние внешнего напряжения питания (зарезервировано для будущих версий)	ВКЛ.: внешнее электропитание имеется ВыКЛ.: внешнее электропитание отсутствует
7	Контроль предохранителя	ВКЛ.: имеется неисправный предохранитель ВыКЛ.: неисправных предохранителей нет
8	не используется	
9	Имеется ошибка малой или средней значимости	ВКЛ.: имеется ошибка малой или средней значимости ВыКЛ.: ошибок малой или средней значимости не имеется
10	Состояние неисправности модуля	00: неисправностей нет      01: незначительная неполадка
11		10: неиспр. средней тяжести    11: серьезная неисправность
12	Состояние готовности модуля	ВКЛ.: нормальный режим ВыКЛ.: в модуле имеется ошибка
13	не используется	
14	Модуль серии "А" или "Q"	ВКЛ.: модуль серии "А" ВыКЛ.: модуль серии "Q"
15	Монтажное состояние модулей	ВКЛ.: модули установлены ВыКЛ.: модули не установлены

### Источники ошибок

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

Значение n1 находится вне диапазона от 0 до FF<sub>H</sub> (код ошибки 4100).

Значение n2 находится вне диапазона от 0 до FF<sub>H</sub> (код ошибки 4100).

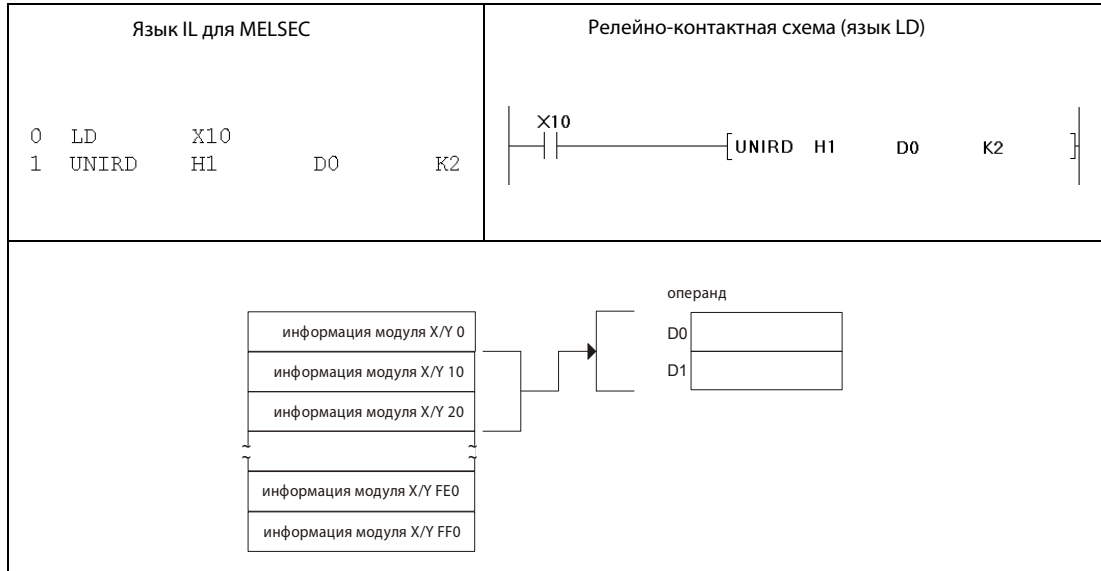
Сумма значений n1 и n2 превышает 256 (код ошибки 4100).



**Пример**

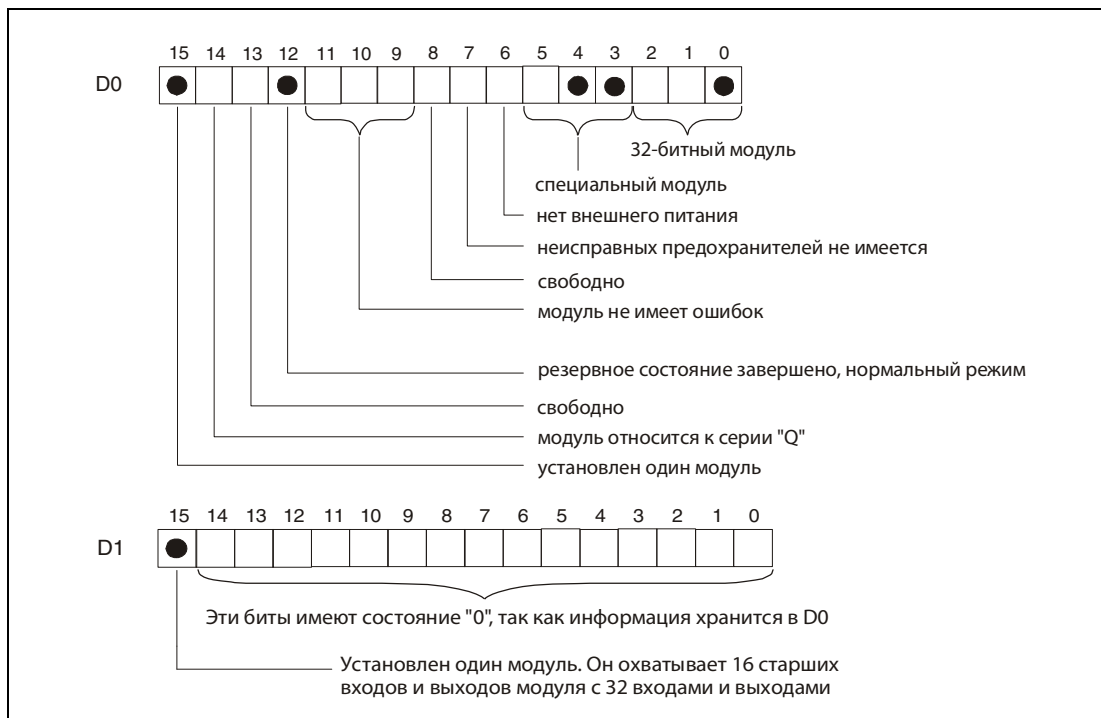
UNIRD (GX Developer)

В следующей программе при включении входа X10 информация модулей, занимающих адреса ввода-вывода с 10<sub>H</sub> по 20<sub>H</sub>, сохраняется в регистрах D0 и D1.



В вышеприведенном примере программы информацию модуля можно считать из регистров D0 и D1. Например, эта информация может быть такой, как это показано на следующем рисунке.

- В случае 32-битного специального модуля серии "System Q". В случае 48-битного или 64-битного модуля содержимое D2 или D2 и D3 идентично содержимому D1.



- В случае 32-битного специального модуля серии "А". В случае 48-битного или 64-битного модуля содержимое D2 или D3 идентично содержимому D1.



- Если разъем (слот) не занят



## 9.2 Поиск и устранение ошибок

### 9.2.1 TRACE, TRACER

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

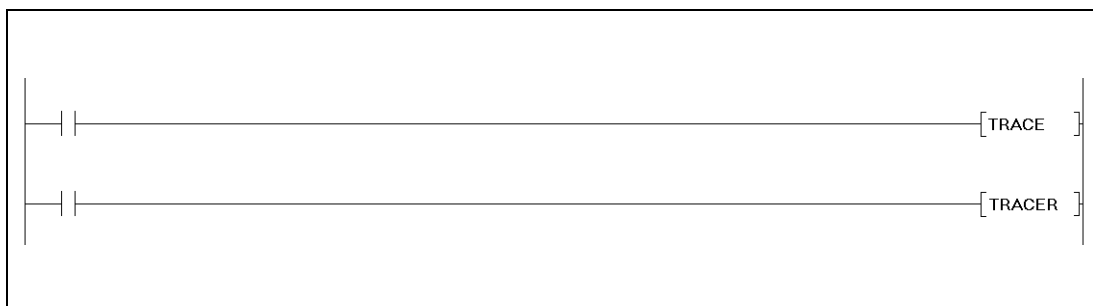
Операнды  
MELSEC Q

Операнды										Флаг ошибки	Шагов
Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы K, H (16#)	Иные			
битовые	словные		битовые	словные							
—	—	—	—	—	—	—	—	—	—	—	1

GX IEC  
Developer



GX  
Developer



Переменные

Операнд	Значение	Тип данных
—	—	—

**Принцип действия**      **Установка и сброс контроля****TRACE      Установка контроля**

Если установлены маркеры SM800, SM801 и SM802, функция трассировки сохраняет данные операндов, выбранных с помощью программатора, в файле трассировки на карте памяти. При выполнении команды TRACE устанавливается маркер SM803. По окончании сбора данных в соответствии с заданиями данные сохраняются и трассировка прекращается.

Если во время трассировки был сброшен маркер SM801, сбор данных прекращается.

После выполнения команды TRACE устанавливается маркер SM805.

Во время трассировки прочие команды TRACE игнорируются. Лишь после выполнения команды TRACER возможно повторное выполнение команды TRACE.

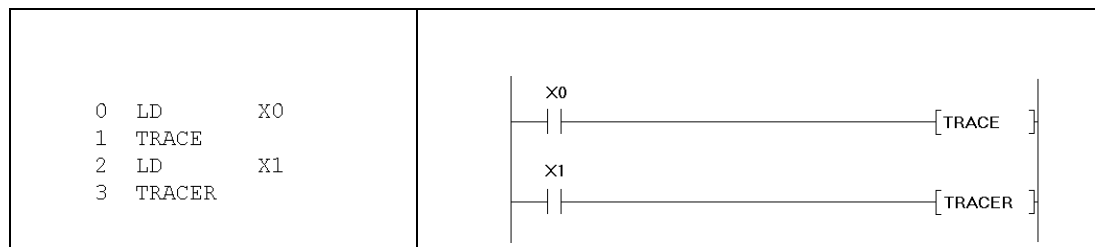
**TRACER      Сброс контроля**

Команда TRACER сбрасывает команду TRACE и маркеры с SM803 по SM805. После завершения выполнения команды TRACER деблокируется повторное выполнение команды TRACE.

**ПРИМЕЧАНИЕ**      *Дополнительная информация на тему трассировки имеется в руководствах по модулям центральных процессоров "System Q", а также в руководствах по средам программирования "GX Developer" и "GX IEC Developer".*

**Пример**      TRACE, TRACER (GX Developer)

В следующей программе при включении входа X0 выполняется команда TRACE. При включении входа X1 команда TRACE сбрасывается с помощью команды TRACER.



## 9.3 Передача данных в файлы и из файлов

### 9.3.1 FWRITE

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					● <sup>1</sup>

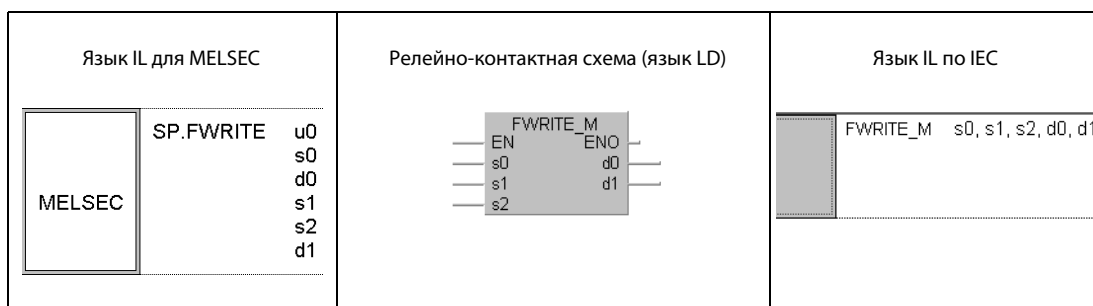
<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

Операнды MELSEC Q

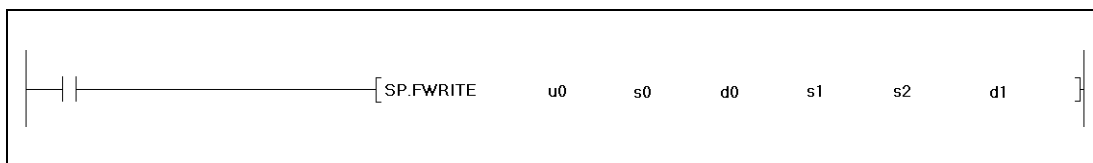
	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s0	●	●	●	—	—	—	—	●	—	SM0	11
d0	—	●	●	—	—	—	—	—	—		
s1	—	●	●	—	—	—	—	—	—		
s2	—	●	●	—	—	—	—	—	●		
d1	●*	●*	●*	—	—	—	—	—	—		

\* Локальные операнды и операнды, зарезервированные для программ, не допускаются.

GX IEC Developer



GX Developer



## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных		
u0	фиктивный (не используется)	—	—	BIN, 16 бит		
s0	Указание дисководов Можно указывать только дисковод с картой памяти ATA (дисковод 2). Карту памяти ROM или обычную память RAM/ROM использовать нельзя.	2	пользователь			
d0	Первый операнд области, в которой сохранены управляющие данные. Необходимы следующие управляющие данные.				BIN, 16 бит	
	Операнд	Значение	Описание	Диапазон		Устанавливает
	(d0)	Режим выполнения	Указывает тип передачи данных: 0000 <sub>H</sub> : передача двоичных данных 0100 <sub>H</sub> : перед передачей данные преобразуются в форма CSV	0000 <sub>H</sub> 0100 <sub>H</sub>		пользователь
	(d0) + 1	резерв	используется системой	—		система
	(d0) + 2	Количество переданных данных	Содержит количество действительно переданных данных. Единица измерения устанавливается в результате указания способа передачи (пословная или побайтовая передача).	—		система
	(d0) + 3	не используется	—	—		—
	(d0) + 4 (d0) + 5	Начальный адрес в целевом файле	Выбор, начиная с какого адреса в целевом файле требуется ввести данные. Двоичные данные (d0 = 0000 <sub>H</sub> ): 00000000 <sub>H</sub> : начало файла 00000001 <sub>H</sub> ...FFFFFFE <sub>H</sub> : начиная с указанного адреса. Единица измерения устанавливается в результате указания способа передачи (пословная или побайтовая передача). FFFFFFF <sub>H</sub> : данные записываются в конец файла Если данные передаются после преобразования из формата CSV (d0 = 0100): В случае центральных процессоров с серийным номером до 01111 в качестве целевого адреса необходимо указать начало файла (00000000) <sub>H</sub> . В случае центральных процессоров, начиная с серийного номера 01112: 00000000 <sub>H</sub> ...FFFFFFE <sub>H</sub> : начало файла FFFFFFF <sub>H</sub> : данные записываются в конец файла.	00000000 <sub>H</sub> ... FFFFFFF <sub>H</sub>		пользователь
	(d0) + 6	Количество столбцов	Количество столбцов, в которые данные записываются в целевой файл в формате CSV. 0: : без указания столбцов >0 : данные записываются в соответств. количество столбцов	от 0 до 65535		пользователь
(d0) + 7	Тип передаваемых данных-словные или байтовые	0: словные 1: байтовые	0, 1	пользователь		

**Переменные**

Операнд	Значение			Диапазон	Устанавливает	Тип данных
s1	Первый операнд области, в которой сохранено название файла.					
	Операнд	Значение	Описание	Диапазон	Устанавливает	BIN, 16 бит
	от (s1) до (s1) + n	Обозначение файла	Обозначение файла состоит из названия файла (максимум 8 знаков), точки и 3-значного расширения (ABD.BIN). Расширение можно не указывать, однако в этом случае не следует указывать и точку. Если указаны более 8 знаков, то расширение игнорируется (даже если оно указано). Расширение BIN или CSV добавляется автоматически.	строковая величина	пользователь	
s2	Первый операнд области, в которой сохранены данные.					
	Операнд	Значение	Описание	Диапазон	Устанавливает	BIN, 16 бит
	(s2)	Длина данных	Указание длины передаваемых данных (количества слов). Даже если в (d0) + 7 выбрана побайтовая передача, количество данных указывается в словах.	от 1 до 480	система	
от (s2) + 1 до (s2) + n	Передаваемые данные	Данные, которые требуется передать.	0000 <sub>H</sub> ... FFFF <sub>H</sub>			
d1	Битовый операнд, устанавливаемый после выполнения команды FWRITE. С помощью (d1) + 1 сигнализируется ошибочное завершение.					
	Операнд	Значение	Описание	Диапазон	Устанавливает	бит
	(d1)	Команда выполнена	Показывает окончание команды FWRITE. Вкл. : команда выполнена Выкл. : команда не выполнена	—	система	
(d1) + 1	Команда выполнена с ошибкой	Показывает, возникли ли ошибки при выполнении команды FWRITE. Вкл. : команда выполнена с ошибкой Выкл.: команда выполнена без ошибок	—			

**ПРИМЕЧАНИЕ** *Среда программирования отображает данные в формате CSV в виде десятичных чисел. Например, буква "A" (41<sub>H</sub>) отображается в виде числа 65. Показываемые значения могут находиться в диапазоне от -32768 до 32767.*

*Допустимые числа для начального адреса в целевом файле при записи двоичных словных данных: диапазон от 00000000<sub>H</sub> до 7FFFFFFF<sub>H</sub> и число FFFFFFFF<sub>H</sub>.*

**Принцип действия**

**Запись данных в указанный файл**

**FWRITE Команда записи**

С помощью команды FWRITE определенное количество данных передается в виде файла на карту памяти ATA. При этом можно выбрать, должны ли данные перед передачей преобразовываться в формат CSV или они передаются без преобразования, в двоичном формате.

При выполнении команды END, если перед этим было распознано завершение команды FWRITE, автоматически устанавливается бит, указанный в (d1) + 0. После команды END в следующем цикле этот бит снова сбрасывается. Этот бит можно использовать в качестве сигнала завершения команды FWRITE.

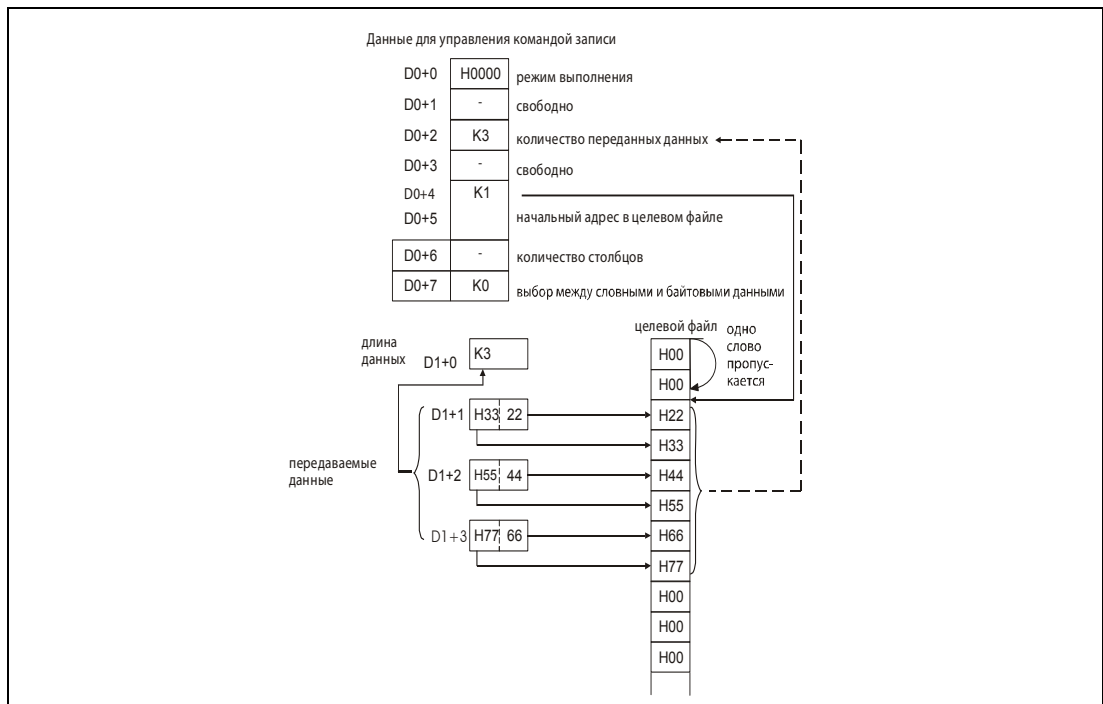
Если при выполнении команды возникла ошибка, то бит, указанный в (d1) + 1, устанавливается и сбрасывается синхронно с сигналом завершения [(d1) + 0].

Пользователь может оценивать этот сигнал в качестве сообщения об ошибке.

Во время выполнения команды FWRITE установлен маркер SM721. Этот маркер устанавливается и при выполнении других команд (S.FREAD, COMRD, PRC). Если маркер SM721 установлен, команда FWRITE не может быть запущена. При такой попытке команда не выполняется.

Если перед выполнением команды (прежде чем устанавливается маркер SM721) распознается ошибка, то сигнал завершения [(d1) + 0], сигнал ошибки [(d1) + 1] и маркер SM721 не устанавливаются.

Длина данных [(s2) + 0] указывается в единицах "слово". На рисунке ниже показана передача двоичных данных.





### Передача двоичных данных

Если передаются двоичные данные и расширение файла не указано, к названию целевого файла присоединяется расширение ".BIN". Если целевой файл не существует, то создается новый файл и данные записываются с начала файла. В качестве атрибутов файла используются архивные атрибуты.

Если длина данных превышает размер целевой области файла, то данные, которые более не умещаются в файл, записываются в конец файла.

Если для сохранения данных указан начальный адрес, превышающий целевой файл, то в случае центральных процессоров с серийным номером до "01111" (первые пять разрядов) возникает сообщение об ошибке. Центральные процессоры с серийным номером, начиная с "01112" в этом случае не передают никакие данные и заканчивают команду без сообщения об ошибке.

Если превышен объем памяти на карте памяти, то прежде чем выработывается сообщение об ошибке, сохраняется столько данных, сколько это возможно. Данные, которые удалось записать, остаются на карте.

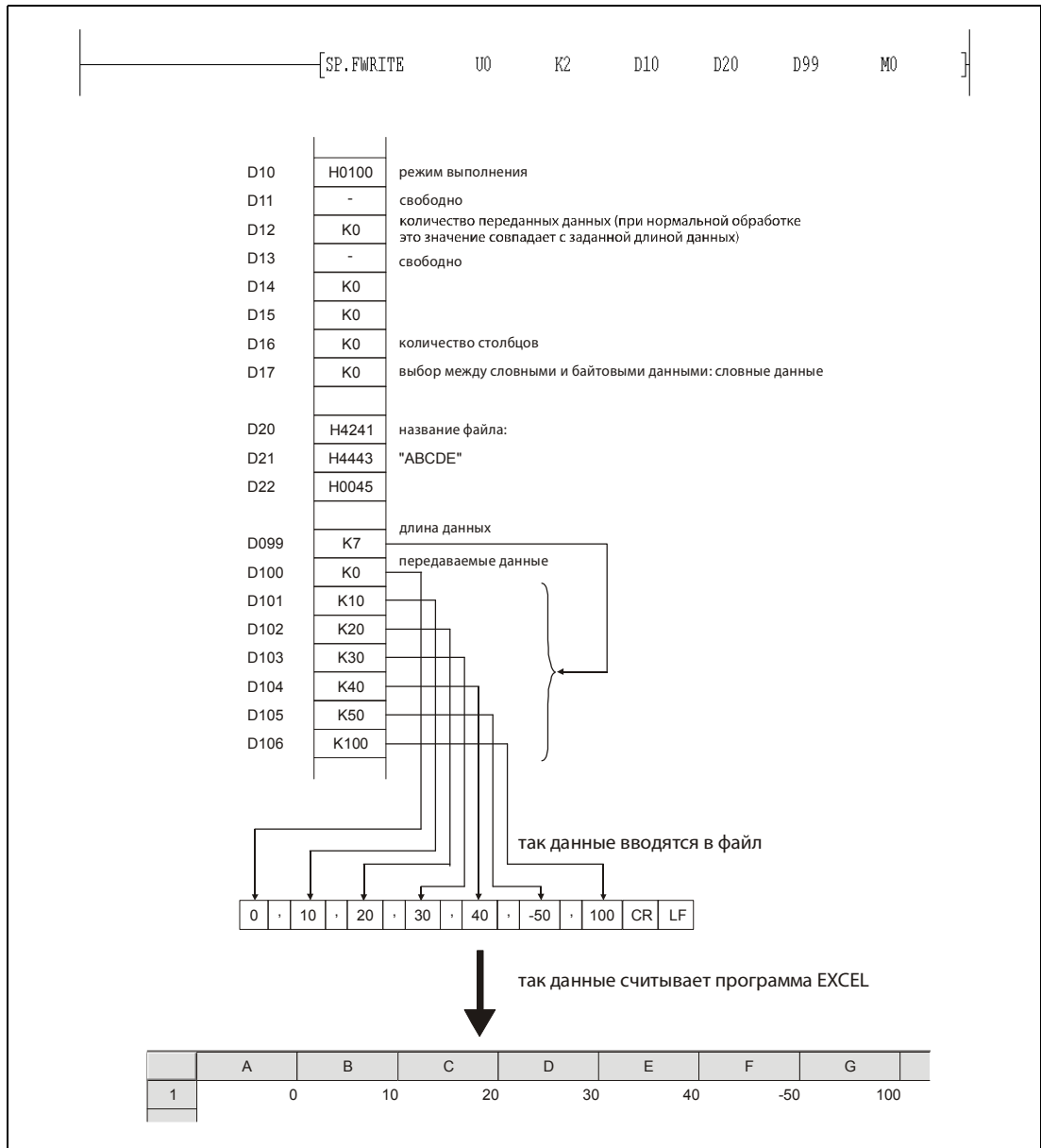
### Передача данных в формате CSV

Если расширение названия файла не указано, то к названию добавляется ".CSV". Если в качестве целевого файла указан уже существующий файл, то в случае центральных процессоров с серийным номером до "01111" (первые пять разрядов) содержимое этого файла стирается, а затем данные записываются с начала файла.

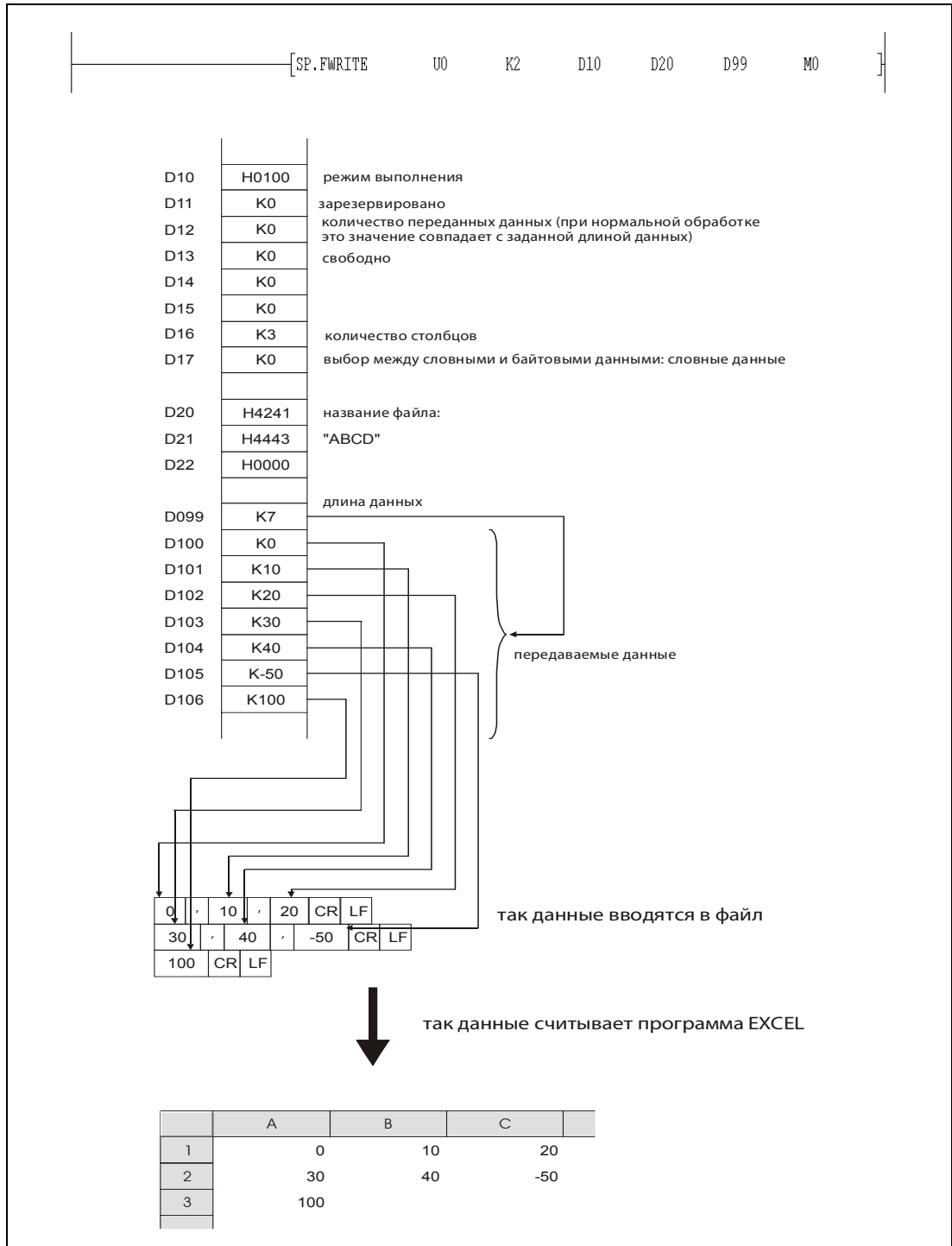
При использовании центральных процессоров с серийным номером "01112" и выше содержимое имеющегося файла стирается и данные записываются с начала файла только в том случае, если в (d0) + 4 и (d0) + 5 указано иное значение кроме FFFFFFFF<sub>H</sub>. Если в (d0) + 4 и (d0) + 5 указано значение FFFFFFFF<sub>H</sub>, данные присоединяются к концу файла.

Если целевой файл не существует, то создается новый файл и данные записываются с начала файла. В качестве атрибутов файла используются архивные атрибуты. Если во время записи превышен объем памяти на карте памяти, то прежде чем выработывается сообщение об ошибке, сохраняется столько данных, сколько это возможно. Данные, которые к этому моменту удалось сохранить, остаются на карте.

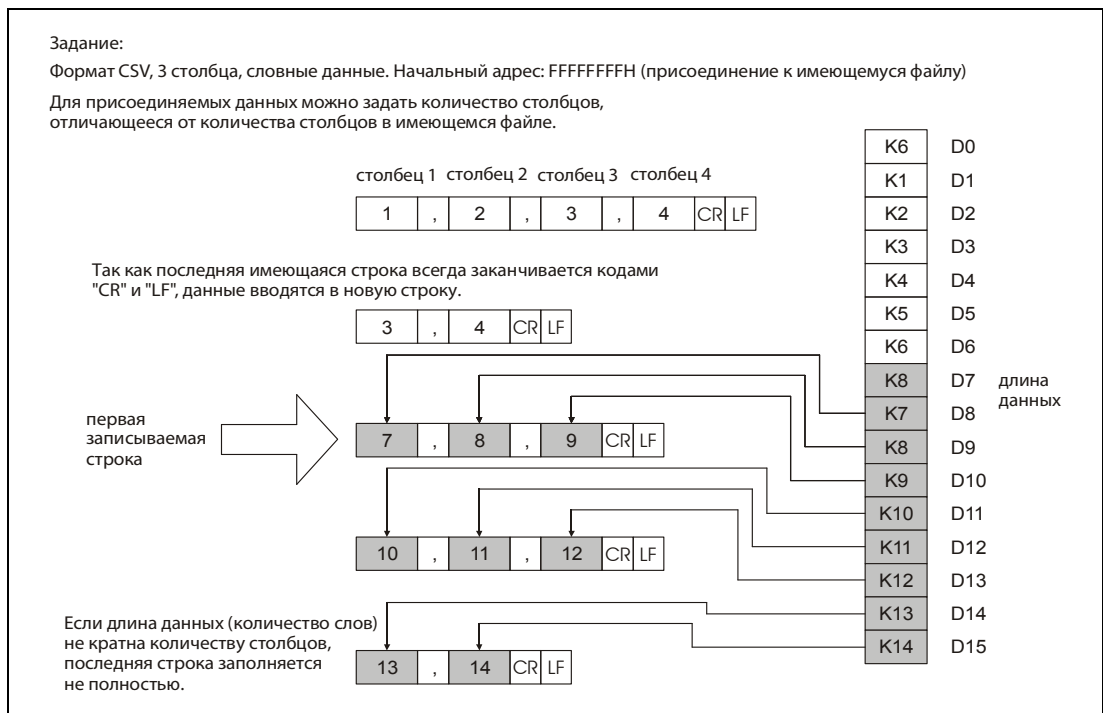
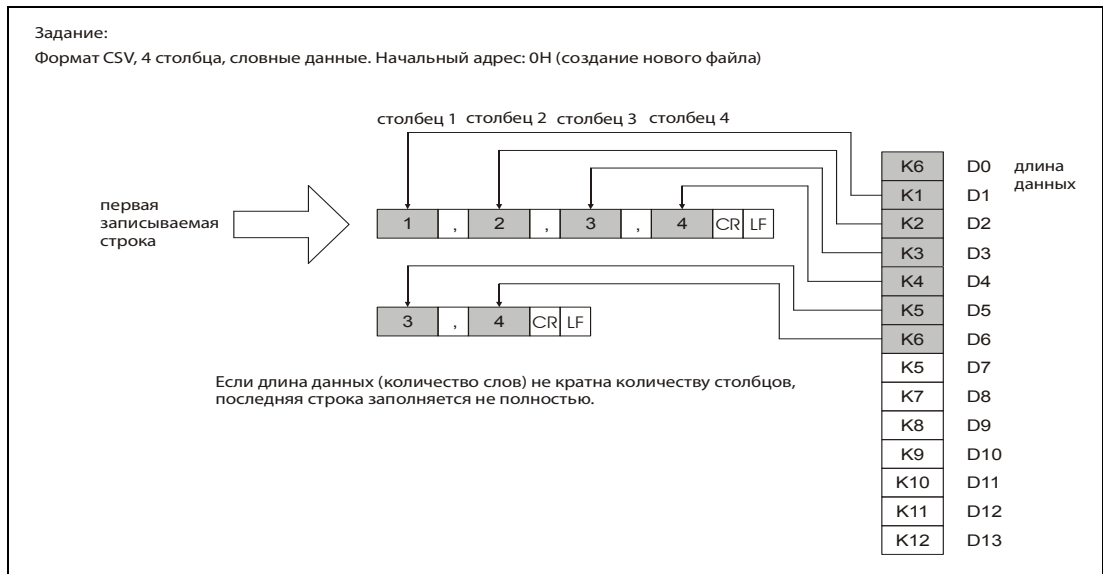
Если в качестве количества записываемых столбцов указан 0, то данные записываются в однострочном формате CSV. Этот случай пояснен на следующей иллюстрации.



Если указанное количество столбцов больше 0, то данные записываются в виде таблицы в формате CSV с соответствующим количеством столбцов. Это показано на следующей иллюстрации.



На двух следующих рисунках приведены примеры записи данных в случае центрального процессора с серийным номером "01112" или выше (первые пять разрядов).



**ПРИМЕЧАНИЕ** Не используйте команду FWRITE в программе прерывания.

**Источники  
ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- В указанном в  $s0$  дисководе находится карта памяти, не являющаяся картой памяти АТА (код ошибки 4100).
- Значения, указанные в качестве управляющих данных, находятся вне допустимых диапазонов (код ошибки 4100).
- Длина данных, указанная в  $(s2) + 0$ , превышает допустимый диапазон либо больше количества данных начиная с  $(s2) + 1$  (код ошибки 4101).
- Превышен объем памяти на карте памяти (код ошибки 4100).
- Указан недействительный операнд (код ошибки 4004).

**Пример 1** FWRITE (GX Developer)

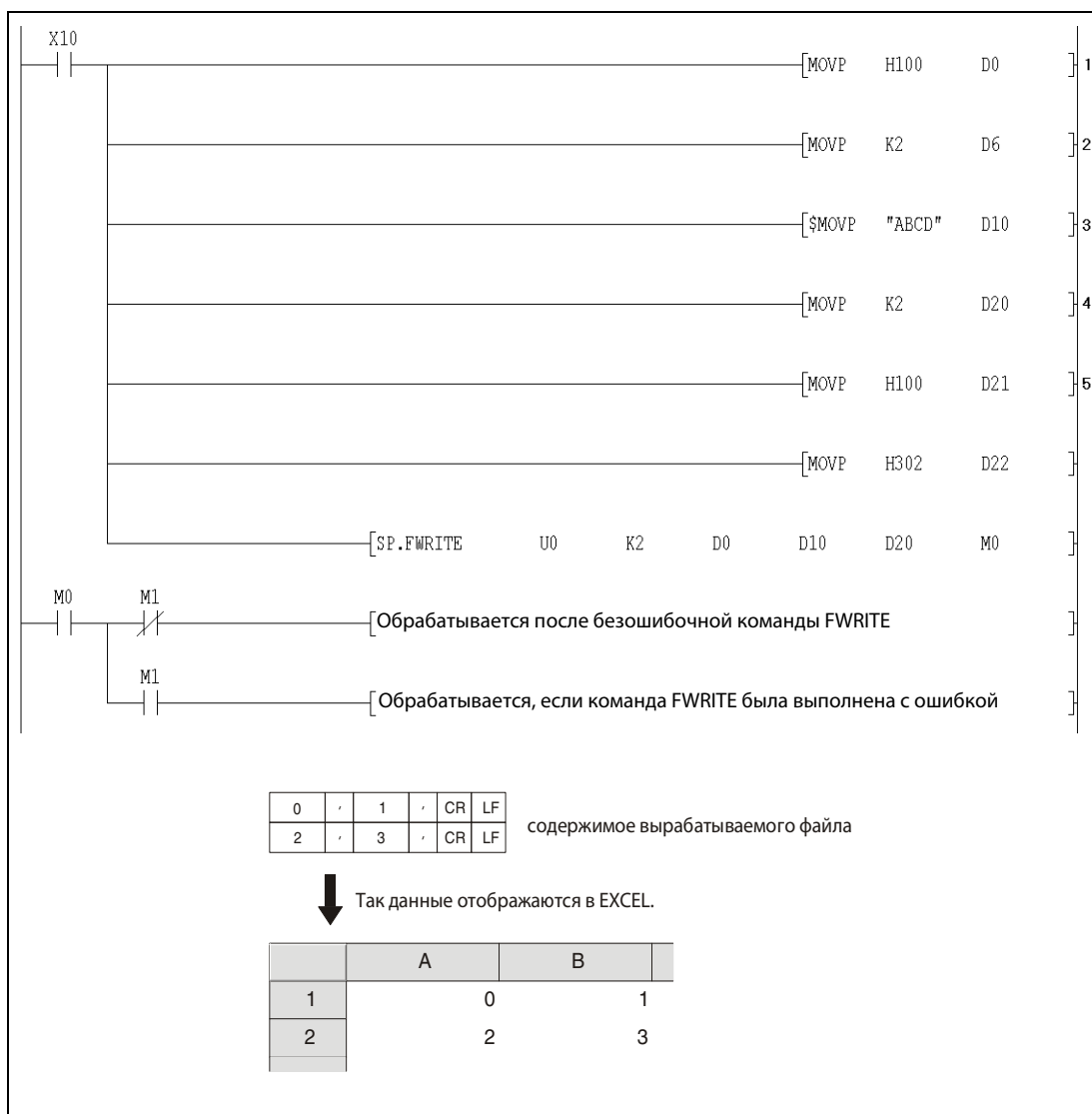
В следующей программе при включении входа X10 четыре байта двоичных данных с содержанием 00<sub>H</sub>, 01<sub>H</sub>, 02<sub>H</sub> и 03<sub>H</sub> присоединяются к файлу "ABCD.BIN". Карта памяти вставлена в дисковод 2. Для управляющих данных занимают восемь операндов, начиная с D0.



- <sup>1</sup> выбор режима выполнения (двоичные данные)
- <sup>2</sup> установка начального адреса (присоединение данных к концу файла)
- <sup>3</sup> ввод названия файла (расширение ".BIN" добавляется автоматически)
- <sup>4</sup> количество передаваемых данных
- <sup>5</sup> указание передаваемых данных

**Пример 2** FWRITE (GX Developer)

При включении входа X10 следующая программа создает файл с названием "ABCD.CSV" на карте памяти в дисковом 2. После этого в этот файл записываются четыре байта с содержимым 00<sub>H</sub>, 01<sub>H</sub>, 02<sub>H</sub> и 03<sub>H</sub> в виде таблицы с двумя столбцами в формате CSV. Для управляющих данных занимают восемь операндов, начиная с D0.



- 1 выбор режима выполнения (формат CSV)
- 2 указание количества столбцов
- 3 ввод названия файла
- 4 количество передаваемых данных
- 5 указание передаваемых данных

### 9.3.2 FREAD

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

**Операнды MELSEC Q**

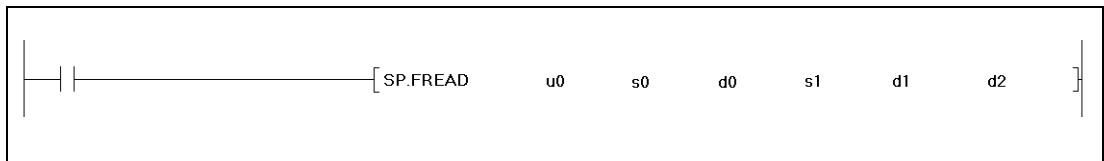
	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□\G□□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s0	●	●	●	—	—	—	—	●	—	SM0	11
d0	—	●	●	—	—	—	—	—	—		
s1	—	●	●	—	—	—	—	—	—		
s2	—	●	●	—	—	—	—	—	●		
d1	●*	●*	●*☆	—	—	—	—	—	—		

\* Локальные операнды и операнды, зарезервированные для программ, не допускаются.

**GX IEC Developer**



**GX Developer**





**Переменные**

Операнд	Значение	Диапазон	Устанавливает	Тип данных		
u0	фиктивный (не используется)	—	—	BIN, 16 бит		
s0	Указание дисководов Можно указывать только дисковод с картой памяти ATA (дисковод 2). Карту памяти ROM или обычную память RAM/ROM использовать нельзя.	2	пользователь			
d0	Первый операнд области, в которой сохранены управляющие данные. Необходимы следующие управляющие данные				BIN, 16 бит	
	Операнд	Значение	Описание	Диапазон		Устанавливает
	(d0)	Режим выполнения	Указывает тип передачи данных: 0000 <sub>H</sub> : передача двоичных данных 0100 <sub>H</sub> : перед передачей данные преобразуются из формата CSV в двоичный формат	0000 <sub>H</sub> 0100 <sub>H</sub>		пользователь
	(d0) + 1	зарезервировано	используется системой	—		система
	(d0) + 2	Количество передаваемых данных	Указание длины передаваемых данных (количества слов). Даже если в (d0) + 7 выбрана побайтовая передача, количество данных указывается в словах.	от 1 до 480		пользователь
	(d0) + 3	не используется	—	—		—
	(d0) + 4 (d0) + 5	Начальный адрес в исходном файле	Выбор, начиная с какого адреса в исходном файле требуется считать данные. Двоичные данные (d0 = 0000): 00000000 <sub>H</sub> : с начала файла 00000001 <sub>H</sub> ...FFFFFFC <sub>H</sub> : начиная с указанного адреса. Единица измерения устанавливается в результате указания способа передачи (пословная или побайтовая передача). FFFFFFFD <sub>H</sub> : не допускается. Если данные передаются после преобразования из формата CSV (d0 = 0100): В случае центральных процессоров с серийным номером до 01111 в качестве начального адреса следует указать начало файла (00000000 <sub>H</sub> ). В случае центральных процессоров с серийным номером 01112 и выше: 00000000 <sub>H</sub> : с начала файла 00000001 <sub>H</sub> ...FFFFFFC <sub>H</sub> : начиная с указанной строки. FFFFFFFD <sub>H</sub> : Считывание продолжается с предшествующей позиции.	00000000 <sub>H</sub> ... FFFFFFFC <sub>H</sub> FFFFFFFD <sub>H</sub>		пользователь
	(d0) + 6	Количество столбцов	Указание количества столбцов в считываемых данных. 0: : данные не разделены на столбцы > 0 : данные сохранены в соответств. количестве столбцов	0, от 1 до 65535		пользователь
(d0) + 7	Тип передаваемых данных-словные или байтовые	0: словные 1: байтовые	0, 1	пользователь		

## Переменные

Операнд	Значение			Диапазон	Устанавливает	Тип данных
s1	Первый операнд области, в которой сохранено название файла					
	Операнд	Значение	Описание	Диапазон	Устанавливает	BIN, 16 бит
	от (s1) до (s1) + n	Обозначение файла	Обозначение файла состоит из названия файла (максимум 8 знаков), точки и трехзначного расширения (ABD.BIN). Расширение можно не указывать, однако в этом случае не следует указывать и точку. Если указаны более 8 знаков, то расширение игнорируется (даже если оно указано). Расширение BIN или CSV добавляется автоматически.	строковая величина	пользователь	
Первый операнд области, в которой сохраняются данные.						
d1	Операнд	Значение	Описание	Диапазон	Устанавливает	BIN, 16 бит
	(d1)	Количество переданных данных	Содержит количество действительно переданных данных. Единица измерения устанавливается в результате указания способа передачи (пословная или побайтовая передача).	от 0 до 480	система	
	от (d1) + 1 до (d1) + n	Передаваемые данные	Данные, которые требуется передать из исходного файла.	0000 <sub>H</sub> ... FFFF <sub>H</sub>		
d2	Битовый операнд, устанавливаемый после выполнения команды FWRITE. С помощью (d2) + 1 сигнализируется ошибочное завершение.					
	Операнд	Значение	Описание	Диапазон	Устанавливает	бит
	(d2)	Команда выполнена	Показывает окончание команды FREAD. Вкл. : команда выполнена. Выкл.: команда не выполнена	—	система	
(d2) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды FREAD. Вкл. : команда выполнена с ошибкой Выкл.: команда выполнена без ошибок	—			

## ПРИМЕЧАНИЕ

Среда программирования отображает данные в формате CSV в виде десятичных чисел. Например, буква "A" (41<sub>H</sub>) отображается в виде числа 65. Показываемые значения могут находиться в диапазоне от -32768 до 32767.

Допустимый диапазон начального адреса в исходном файле при считывании словных данных: от 00000000<sub>H</sub> до 7FFFFFFF<sub>H</sub>.

**Принцип действия**

**Считывание данных из указанного файла**

**FREAD Команда считывания**

Команда FREAD считывает определенное количество данных из файла на карте памяти ATA. При этом можно выбрать, должны ли данные перед передачей преобразовываться из формата CSV в двоичный формат или они могут быть считаны непосредственно в двоичном формате.

При выполнении команды END, если перед этим было распознано завершение команды FREAD, автоматически устанавливается бит, указанный в (d2) + 0. После команды END в следующем цикле этот бит снова сбрасывается. Этот бит можно использовать в качестве сигнала завершения команды FREAD.

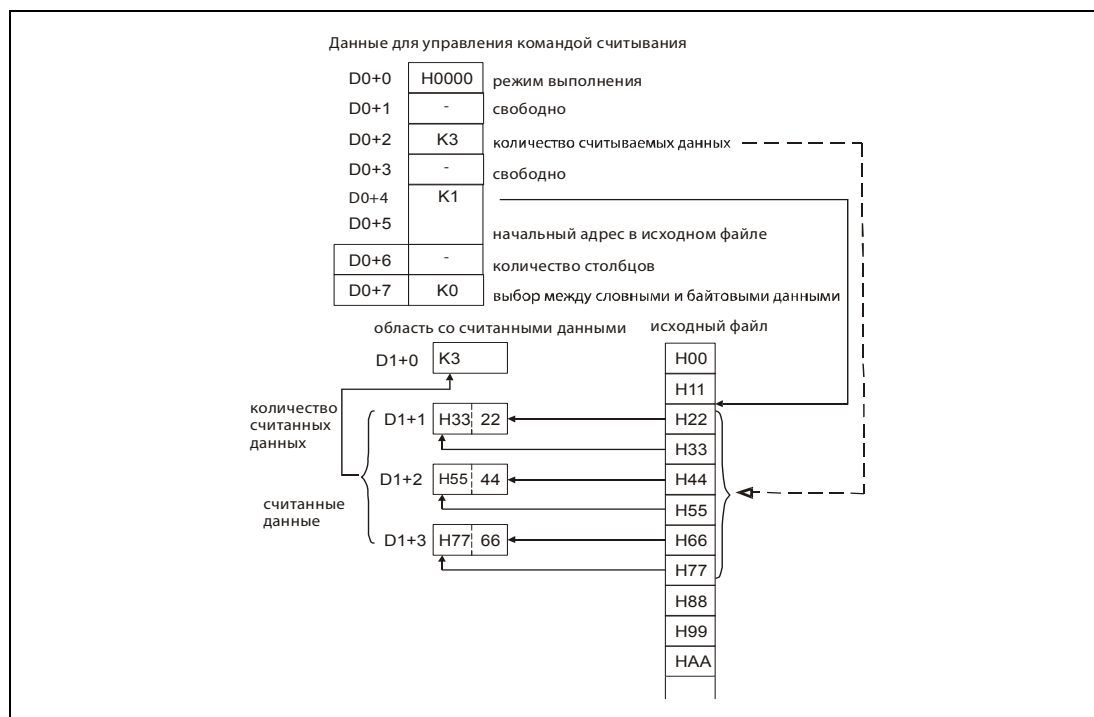
Если при выполнении команды возникла ошибка, то синхронно с сигналом завершения [(d2) + 0] устанавливается и сбрасывается бит, указанный (d2) + 1.

Пользователь может оценивать этот сигнал в качестве сообщения об ошибке.

Во время выполнения команды FREAD установлен маркер SM721. Этот маркер устанавливается и при выполнении других команд (S.WRITE, COMRD, PRC). Если маркер SM721 установлен, команда FREAD не может быть запущена. При такой попытке команда не выполняется.

Если перед выполнением команд (прежде чем был установлен SM721) распознана ошибка, то сигнал о завершении [(d2) + 0], сигнал об ошибке [(d2) + 1] и маркер SM721 не устанавливаются.

Длина данных [(d0) + 0] указывается в "словах". На рисунке ниже показана передача двоичных данных.



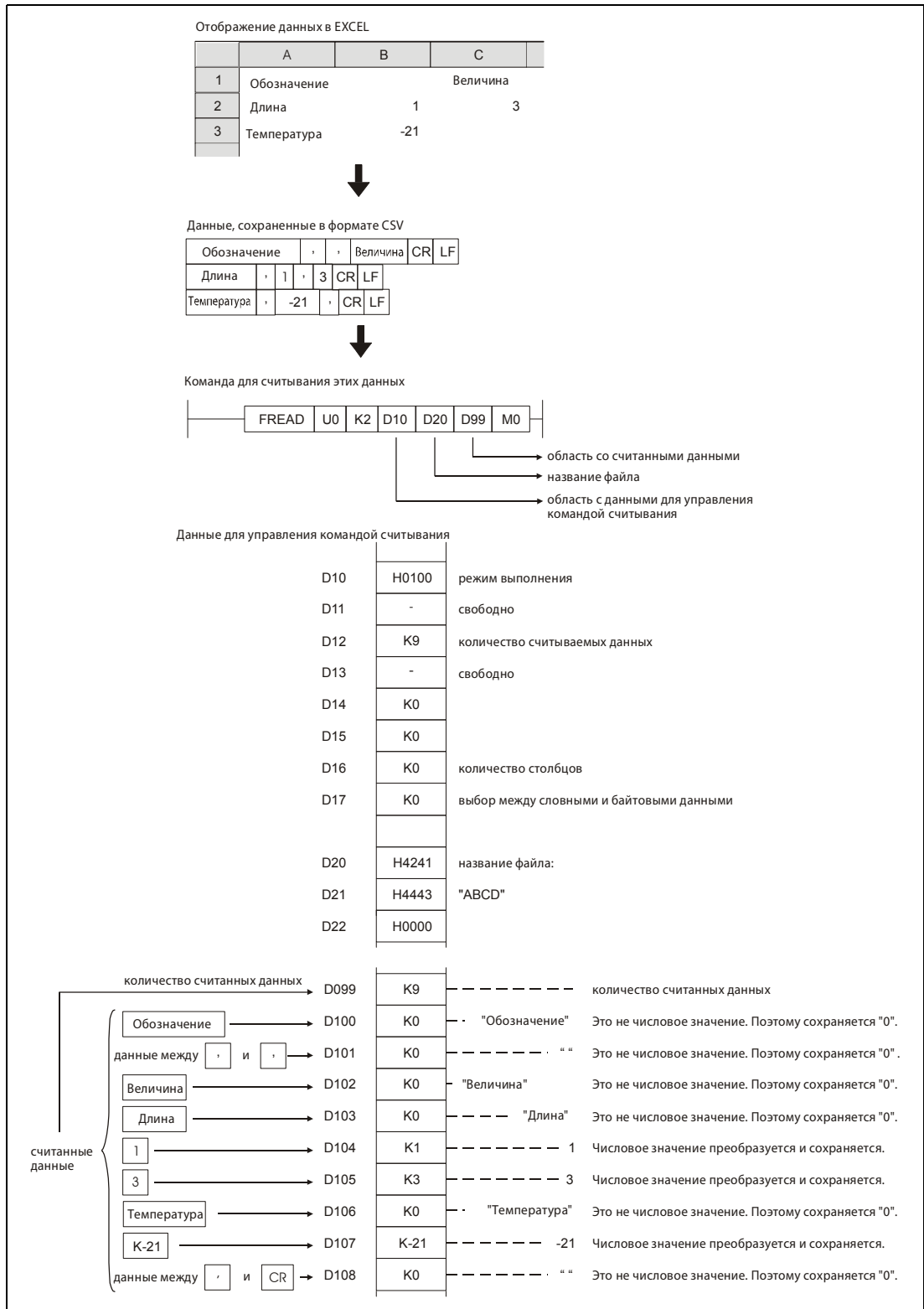
**Передача двоичных данных:**

Если передаются двоичные данные, то к названию исходного файла присоединяется расширение ".BIN". Если такой файл не существует, выводится сообщение об ошибке. Если указанный начальный адрес превышает область исходного файла, то в центральных процессорах с серийным номером до "01111" (первые пять разрядов) выводится сообщение об ошибке. Центральные процессоры с серийным номером, начиная с "01112", в этом случае не передают никакие данные и заканчивают команду без сообщения об ошибке.

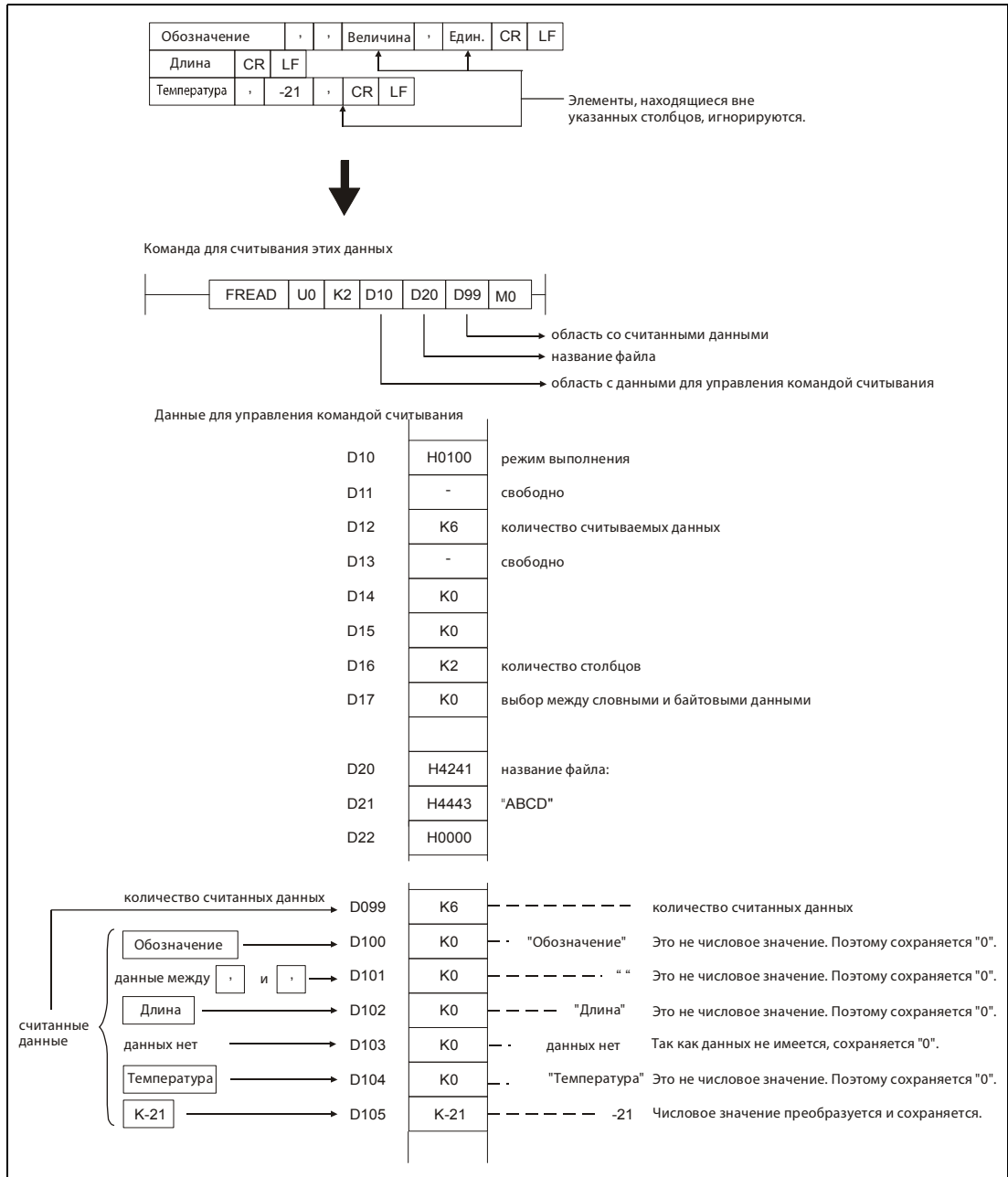
**Передача данных после преобразования из формата CSV:**

Элементы файла в формате CSV (ячейки для программы EXCEL) считываются строка за строкой. Числовые значения преобразуются в двоичные данные и сохраняются в указанных операндах. Если название исходного файла указано без расширения, к нему добавляется ".CSV". Если исходный файл не существует, выводится сообщение об ошибке.

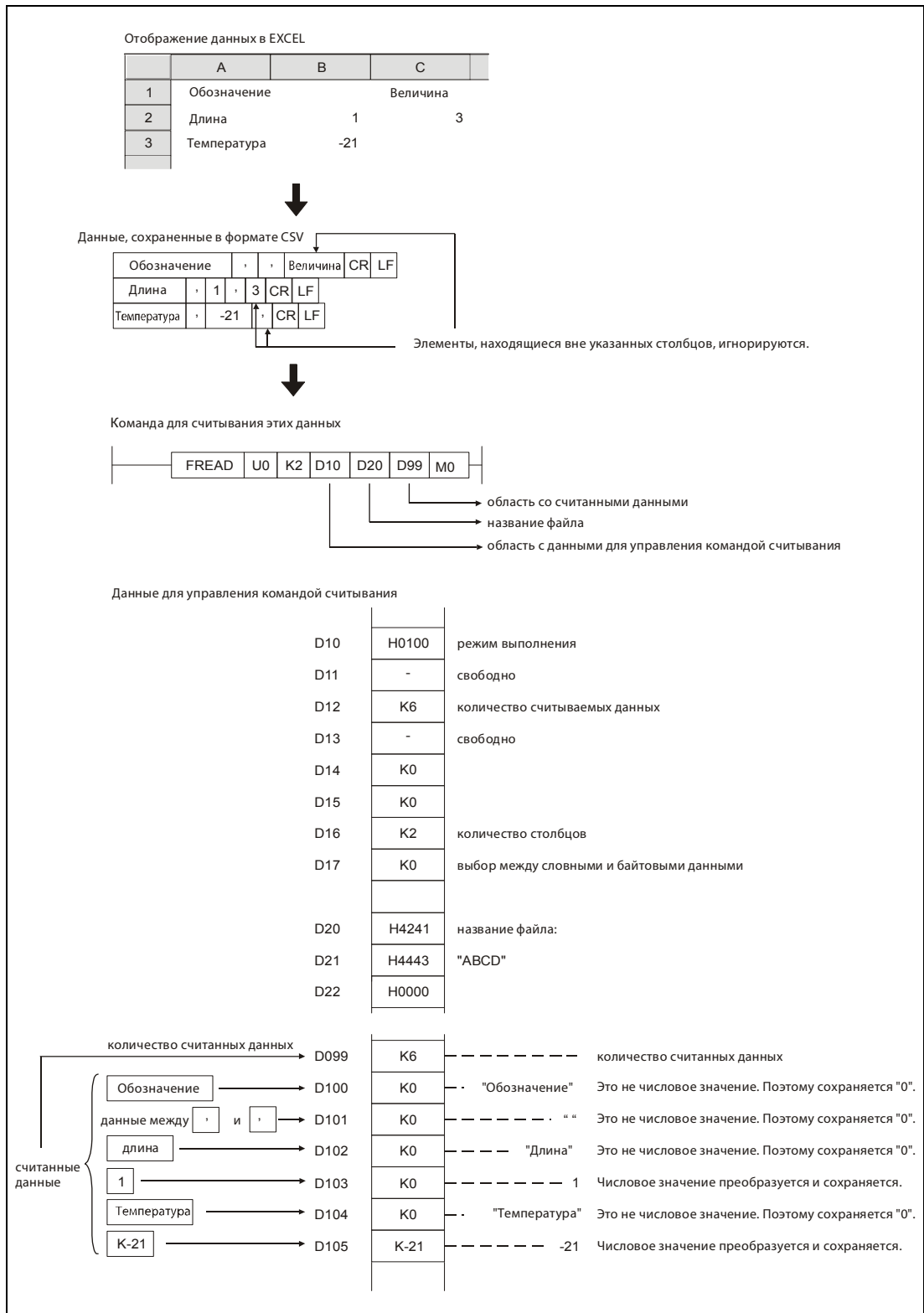
С начала файла считываются столько элементов, сколько было указано в  $[(d0) + 2]$ . Если до того, как было считано указанное количество, достигнут конец файла, центральные процессоры с серийным номером до "01111" (первые пять разрядов) выводят сообщение об ошибке. Процессоры с серийным номером "01112" и выше в этом случае считывают имеющиеся данные. Если для количества столбцов указан 0, то при считывании файла типа CSV игнорируется разделение на строки. Это пояснено иллюстрацией на следующей странице.



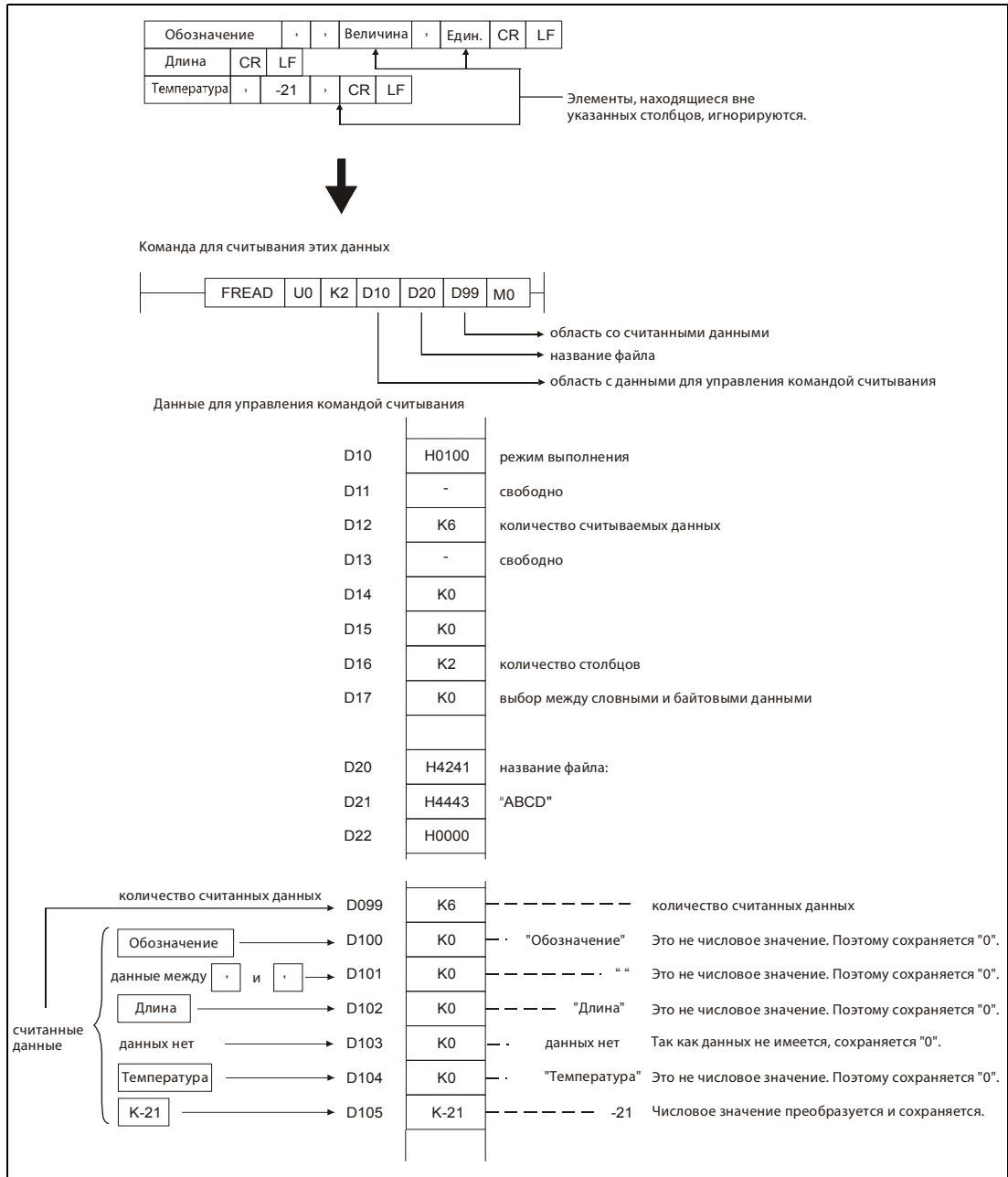
Если количество столбцов в каждой строке изменяется, то данные также считываются без соблюдения строк. Программа EXCEL такие файлы не создает, однако они могут образоваться после вмешательства пользователя.



Если выбрано считывание данных после преобразования из формата CSV и заданное количество столбцов не равно 0, то ожидается, что в файле типа CSV имеется таблица данных с соответствующим количеством столбцов. Элементы, находящиеся вне этих столбцов, игнорируются. Этот случай показан на следующей иллюстрации.

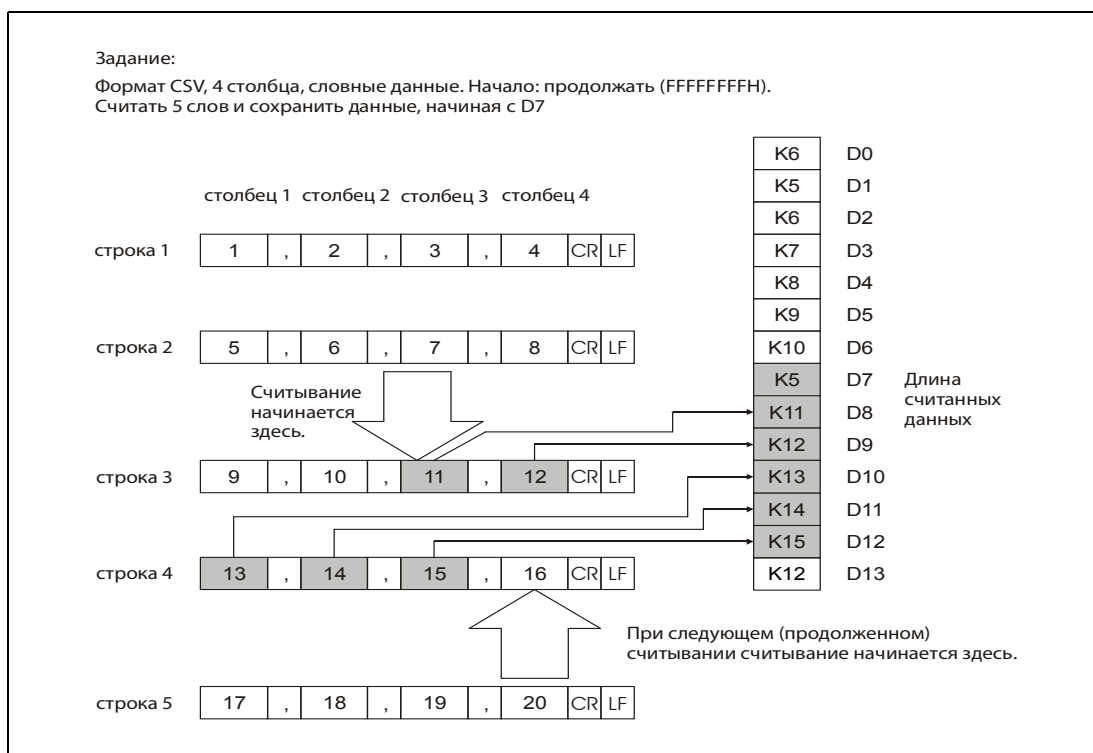
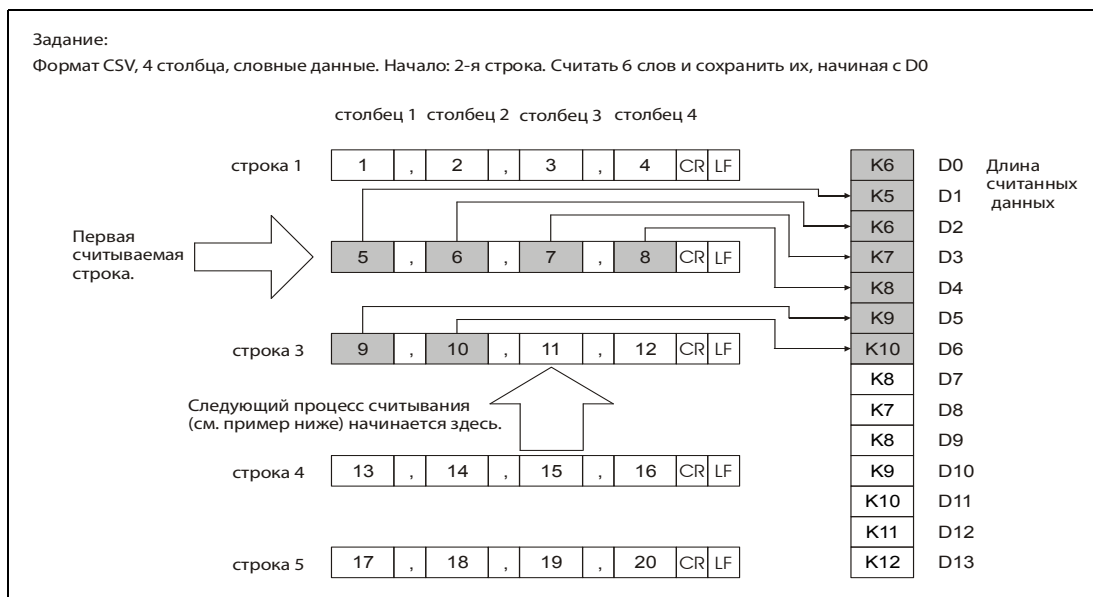


Если количество столбцов в каждой строке изменяется, но было указано определенное количество столбцов, то элементы, находящиеся вне указанных столбцов, игнорируются, а в случае, если элементы отсутствуют, вместо них считывается "0".  
 Если фактическое число строк меньше, чем это указано в количестве передаваемых данных [(d0) + 2], то отсутствующие строки считываются в виде нулей.





На двух следующих иллюстрациях показан пример двукратного считывания данных из одного и того же файла при использовании центрального процессора с серийным номером "01112" (первые пять разрядов) или выше.



При продолжении считывания данных из того же файла, исходя из позиции предшествующего считывания, между отдельными процессами считывания нельзя изменять настройки формата данных, количества столбцов и типа данных (байтовые или словные).

Во время продолженного считывания нельзя выполнять никакие другие команды FREAD или FWRITE.

Числовые значения формата CSV преобразуются, как это показано в следующей таблице.

Числовые значения в формате CSV	Словный операнд	
	без арифметического знака	с арифметическим знаком
-32768	32768	-32768
-1	65535	-1
0	0	0
1	1	1
32767	32767	32767
32768	32768	-32768
65535	65535	-1

Вместо числовых значений, находящихся вне допустимого диапазона, а также вместо элементов, для которых числовые значения отсутствуют, при преобразовании из формата CSV принимается значение 0.

#### Источники ошибок

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

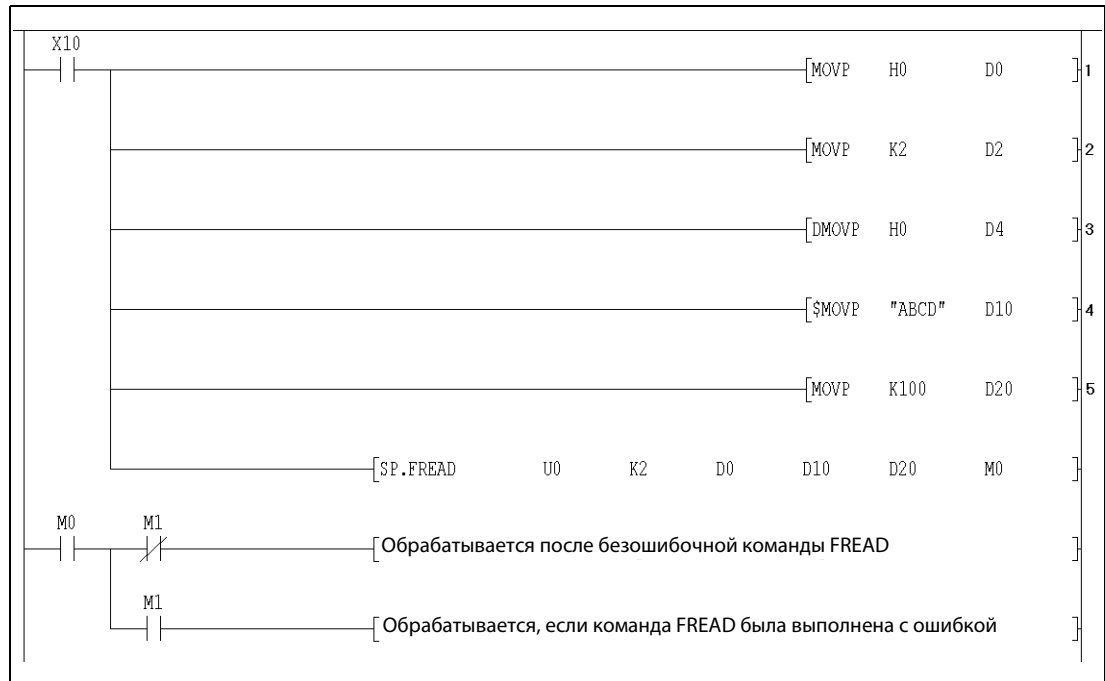
- В указанном в s0 дисковом находится карта памяти, не являющаяся картой памяти АТА (код ошибки 4100).
- Значения, указанные в качестве управляющих данных, находятся вне допустимых диапазонов (код ошибки 4100).
- Длина данных, указанная в (d0) + 0, превышает допустимый диапазон (код ошибки 4101).
- Указан недействительный операнд (код ошибки 4004).
- Файл с названием, указанным в s1, на указанном дисковом не существует (код ошибки 2410).
- Длина считываемых данных превышает размер целевой области (код ошибки 4101).
- При считывании двоичных данных длина данных в исходном файле меньше количества данных, заданного начиная с (d0) + 2 (код ошибки 4100).

**Пример 1** FREAD

В следующей программе при включении входа X10 из файла "ABCD.BIN" считываются четыре байта двоичных данных. Считывание начинается с начала файла. Файл "ABCD.BIN" хранится на карте памяти, вставленной в дисковод 2.

Для управляющих данных занимают восемь операндов, начиная с D0.

Для считанных данных зарезервированы сто байтов, начиная с D20.



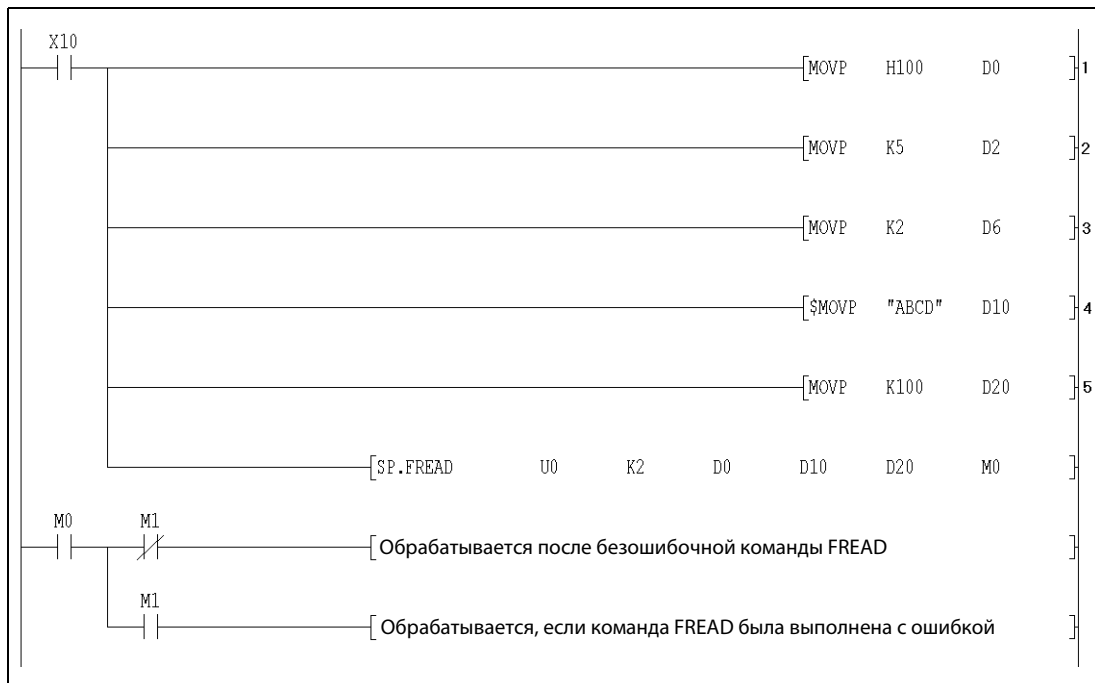
- 1 выбор режима выполнения (двоичные данные)
- 2 количество считываемых слов данных
- 3 начальный адрес в исходном файле (с начала файла)
- 4 ввод названия файла
- 5 указание размера области для считанных данных

**Пример 2** FREAD (GX Developer)

Следующая программа при включении входа X10 считывает данные из файла "ABCD.CSV", который находится на карте памяти, вставленной в дисковод 2. Данные состоят только из числовых значений, упорядоченных в виде двух столбцов в формате CSV.

Для управляющих данных занимают восемь операндов, начиная с D0.

Для считанных данных зарезервированы сто байтов, начиная с D20.



- 1 выбор режима выполнения (формат CSV)
- 2 количество считываемых слов данных
- 3 указание количества столбцов
- 4 ввод названия файла
- 5 указание размера области для считанных данных

## 9.4 Команды обращения с программой

### 9.4.1 PLOADP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					● <sup>1</sup>

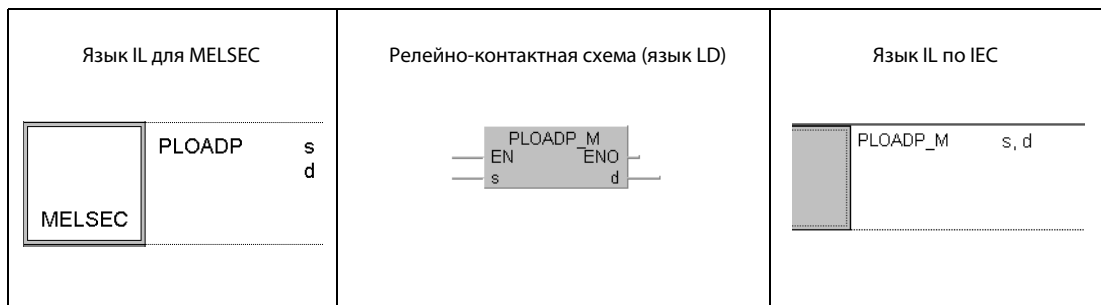
<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

Операнды MELSEC Q

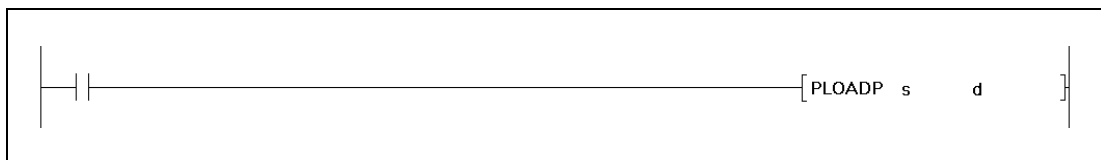
	Операнды							Флаг ошибки	Шагов		
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□NG□	Индексные регистры Zn			Константы K, H (16#)	Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	●	SM0	3
d	●*	—	—	—	—	—	—	—	—		

\* Локальные операнды не допускаются.

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
s	Номер дисковогода, в котором хранится программа, и строковая величина с названием файла или начальный адрес операнда, в котором в виде строковой величины сохранено название файла.	BIN, 16 бит
d	Операнд, устанавливаемый на один цикл после выполнения команды.	бит

**ПРИМЕЧАНИЕ** *Среда GX IEC Developer файловую систему не поддерживает.*

**Принцип действия****Загрузка программы с карты памяти****PLOADP Загрузка программы**

Команда PLOADP загружает программу с карты памяти (которая может быть вставлена в дисковод 1, 2 или 4) во внутреннюю память (дисковод 0) и переключает ее в резервный режим. На дисководе 0 должно иметься достаточно места.

Указывать расширение ".QPG" в названии программного файла не требуется.

Указанный в d операнд устанавливается после обработки команды END того цикла, в котором команда PLOADP была выполнена полностью. После выполнения следующей команды END этот операнд снова выключается.

Команды PLOADP, PUNLOADP и PSWAPP не могут выполняться одновременно. Если одна из этих команд уже выполнена, то какая-либо другая из этих команд, вызванная позднее, не обрабатывается. В программе следует предусмотреть соответствующие блокировки.

Загруженной программе присваивается самый низкий свободный номер программы, имеющийся в центральном процессоре. Номера программ можно выяснить с помощью программатора. Имеется возможность принудительно присвоить загруженной программе определенный номер, записав его в регистр SD720.

Команда PLOADP не может использоваться в программе прерываний.

Для выполнения программы, загруженной с помощью команды PLOADP, используется команда PSCAN.

Все настройки загруженной программы, касающиеся регистров файлов, начальных значений операндов, комментариев и локальных операндов, установлены на "Use PLC file setting" ("использовать настройки файлов, установленные в контроллере").

В качестве настроек обновления входов и выходов для загруженной программы указывается "Disabled" ("дезактивировано").

Во время выполнения команды PLOADP, изменения в программе, сделанные с помощью программатора во время режима "RUN", не передаются. Изменения передаются лишь после выполнения команды PLOADP. И наоборот, во время передачи изменений программы из программатора команда PLOADP не выполняется.

**Источники ошибок**

В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

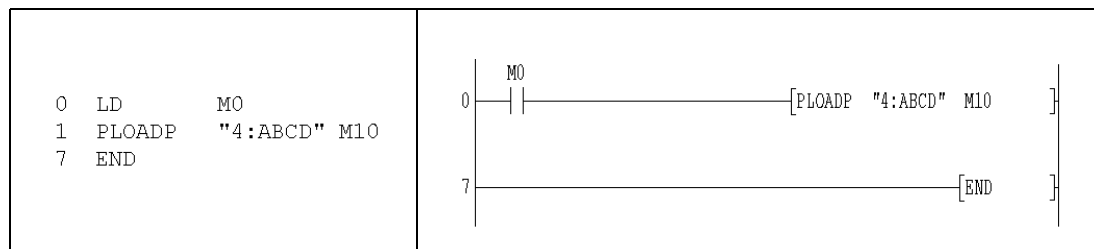
- Программа с указанным названием на указанном дисковом не существует (код ошибки 2410).
- Указан недействительный номер дисковода (код ошибки 4100).
- Во внутренней памяти (дисковод 0) не хватает места для загрузки указанной программы (код ошибки 2413).
- В память для хранения программ уже загружено максимальное количество программ (код ошибки 4101).
- Программа с номером, указанным в SD720, уже существует, или указанный в SD720 номер превышает максимально возможный номер программы (номер ошибки 4101).
- Уже существует программный файл с таким же названием, как у загружаемой программы (номер ошибки 2410).
- Невозможно зарезервировать память для локальных операндов (номер ошибки 2401).

Тип центрального процессора	Размер памяти для программ (количество файлов)	Самый большой номер программы
Q02H	28	28
Q06H	60	60
Q12H	124	124
Q25H	124	124

**Пример**

PLOADP (GX Developer)

В следующем примере при установке маркера M0 программа "ABCD.QPG" из дисковода 4 передается в дисковод 0 и переключается в резервный режим.



### 9.4.2 PUNLOADP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□\G□□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	●	SM0	3
d	●*	—	—	—	—	—	—	—	—		

\* Локальные операнды не допускаются.

GX IEC  
Developer



GX  
Developer



Переменные

Операнд	Значение	Тип данных
s	Строковая величина с названием файла, который требуется стереть, или начальный адрес операнда, в котором в виде строковой величины сохранено название файла.	BIN, 16 бит
d	Операнд, устанавливаемый на один цикл после выполнения команды.	бит



**Принцип действия****Стирание программы из памяти для хранения программ****PUNLOADP Стирание программы**

Команда PUNLOADP стирает программу, загруженную в память для хранения программ (диск-вод 0) и находящуюся в резервном режиме. Резервная программа, к которой обращается команда PSCAN, не может быть стерта.

Указывать расширение ".QPG" в названии программного файла не требуется.

Указанный в d операнд устанавливается после обработки команды END того цикла, в котором команда PUNLOADP была выполнена полностью. После выполнения следующей команды END этот операнд снова выключается.

Команды PUNLOADP, PLOADP и PSWAPP не могут выполняться одновременно. Если одна из этих команд уже выполнена, то какая-либо другая из этих команд, вызванная позднее, не обрабатывается. В программе следует предусмотреть соответствующие блокировки.

Если после стирания программы выключается и снова включается блок питания или выполняется сброс центрального процессора, то вырабатывается сообщение об ошибке "Неправильная настройка программы" с кодом ошибки 2400. Для устранения этой проблемы следует стереть название стертой программы из программных настроек.

Команда PUNLOADP не может использоваться в программе прерываний.

При стирании программы с помощью команды PUNLOADP эта программа должна быть переведена в резервный режим с помощью команды PSTOP.

Во время выполнения команды PUNLOADP изменения в программе, сделанные с помощью программатора во время режима "RUN", не передаются. Изменения передаются лишь после выполнения команды PUNLOADP. И наоборот, во время передачи изменений из программатора команда PUNLOADP не выполняется.

**Источники ошибок**

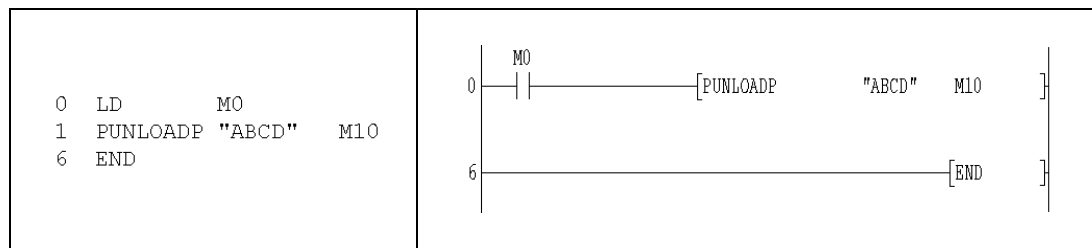
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Программа с указанным названием не существует (код ошибки 2410).
- Программа с указанным названием не находится в резервном режиме или выполняется (код ошибки 4101).
- В памяти для хранения программ имеется только указанная программа (код ошибки 4101).

**Пример**

PUNLOADP (GX Developer)

В следующем примере при установке маркера M0 стирается программа "ABCD.QPG" в диск-воде 0.



### 9.4.3 PSWAPP

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					● <sup>1</sup>

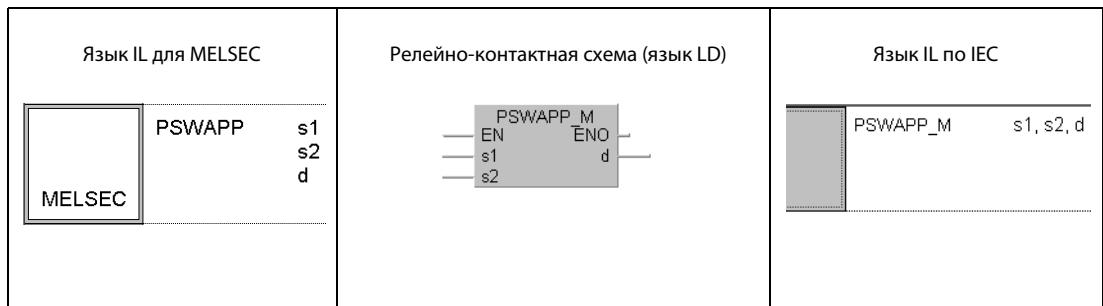
<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

**Операнды MELSEC Q**

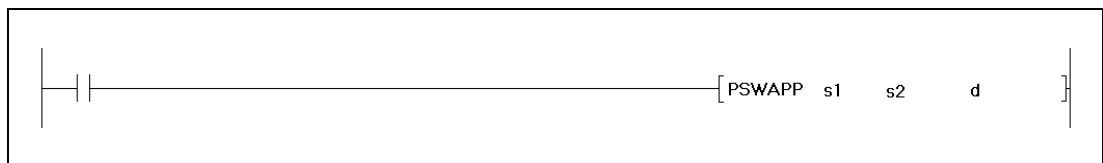
	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	●	SM0	3
s2	—	●	●	—	—	—	—	—	●		
d	●*	—	—	—	—	—	—	—	—		

\* Локальные операнды не допускаются.

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s1	Строковая величина с названием файла, который требуется стереть, или начальный адрес операнда, в котором в виде строковой величины сохранено название файла.	BIN, 16 бит
s2	Номер дисководов, на котором находится загружаемая программа, и строковая величина с названием файла или начальный адрес операнда, в котором в виде строковой величины сохранено название файла.	BIN, 16 бит
d	Операнд, устанавливаемый на один цикл после выполнения команды.	бит

**Принцип действия****Стирание программы из памяти для хранения программ и загрузка программы****PSWAPP Стирание программы и загрузка другой программы**

Команда PSWAPP стирает программу, находящуюся в резервном режиме в памяти для хранения программ (дисковод 0). Резервная программа, к которой обращается команда PSCAN, не может быть стерта.

После этого во внутреннюю память (дисковод 0) передается программа из дисковода 1, 2 или 4 и эта программа переключается в резервный режим. На дисководе 0 должно иметься достаточно места.

Указывать расширение ".QPG" в названии программного файла не требуется.

Указанный в d операнд устанавливается после обработки команды END того цикла, в котором команда PSWAPP была выполнена полностью. После выполнения следующей команды END этот операнд снова выключается.

Загруженной программе присваивается номер стертой программы.

Команды PUNLOADP, PLOADP и PSWAPP не могут выполняться одновременно. Если одна из этих команд уже выполнена, то какая-либо другая из этих команд, вызванная позднее, не обрабатывается. В программе следует предусмотреть соответствующие блокировки.

Если после стирания программы выключается и снова включается блок питания или выполняется сброс центрального процессора, то вырабатывается сообщение об ошибке "Неправильная настройка программы" с кодом ошибки 2400. Для устранения этой проблемы можно в программных настройках заменить название стертой программы названием загруженной программы.

Команда PSWAPP не может использоваться в программе прерываний.

Все настройки загруженной программы, касающиеся регистра файлов, начальных значений операндов, комментариев и локальных операндов, устанавливаются в состояние "Use PLC file setting" ("использовать настройки файлов, установленные в контроллере").

В качестве настроек обновления входов и выходов для загруженной программы указывается "Disabled" ("дезактивировано").

Во время выполнения команды PSWAPP изменения в программе, сделанные с помощью программатора во время режима "RUN", не передаются. Изменения передаются лишь после выполнения команды PSWAPP. И наоборот, во время передачи изменений программы из программатора команда PSWAPP не выполняется.

**Источники ошибок**

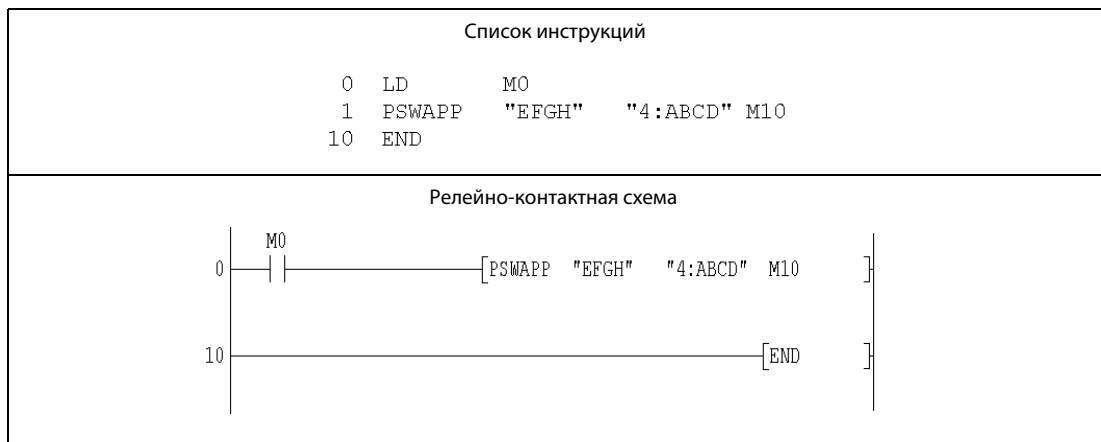
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Дисковод или программа с указанным названием (s1 или s2) не существует (код ошибки 2410).
- Указан недействительный номер дисковода (s2) (код ошибки 4100).
- Во внутренней памяти (дисковод 0) не хватает места для загрузки указанной программы (код ошибки 2413).
- Программа с указанным названием (s1) не находится в резервном режиме или выполняется (код ошибки 4101).

**Пример**

PSWAPP (GX Developer)

В следующем примере при установке маркера M0 стирается программа "EFGH.QPG" в дисковом 0. После этого программа "ABCD.QPG" передается из дисковода 4 в дисковод 0 и переключается в резервный режим.



## 9.5 Команды передачи

### 9.5.1 RBMOV, RBMOV P

Процессор

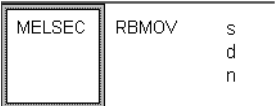
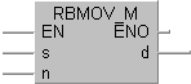
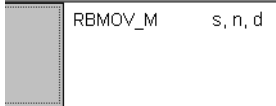
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					● <sup>1</sup>

<sup>1</sup> кроме процессоров Q00J, Q00 и Q01

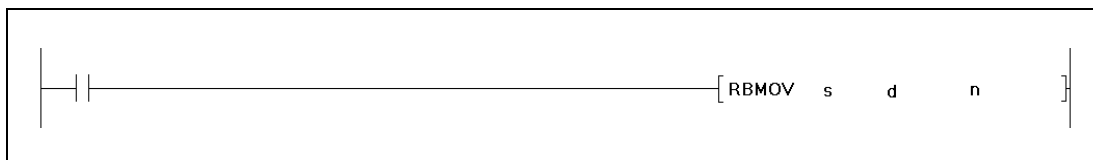
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	●	●	●	●	●	—	—	—	SMO	4	
d	●	●	●	●	●	—	—	—			
n	●	●	●	●	●	●	●	—			

GX IEC Developer

Язык IL для MELSEC	Релейно-контактная схема (язык LD)	Язык IL по IEC
		

GX Developer



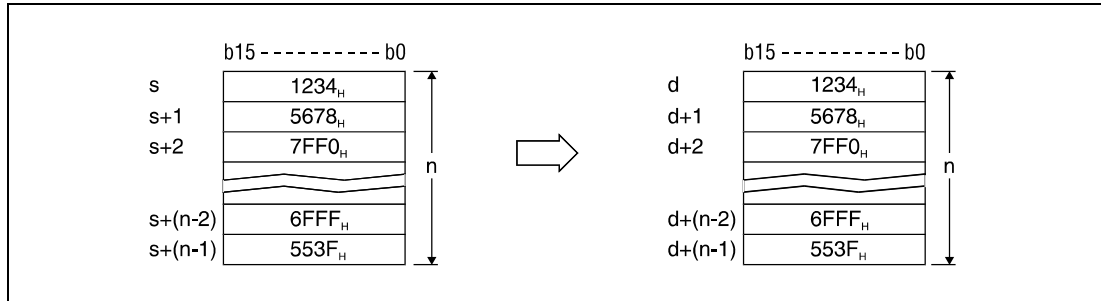
Переменные

Операнд	Значение	Тип данных
s	Начальный адрес операнда, в котором сохранены передаваемые данные.	BIN, 16 бит
d	Начальный адрес операнда, в котором сохраняются данные.	
n	Количество передаваемых блоков данных	

**Принцип действия**

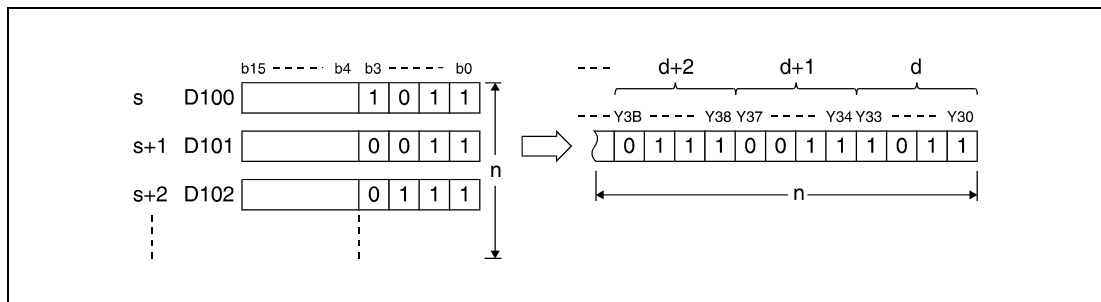
**Передача блоков двоичных данных с высокой скоростью**  
**RBMOV/RBMOV P Поблочная передача данных (16 бит)**

Команда RBMOV позволяет за один раз передать блок следующих друг за другом адресов. В *s* устанавливается первый передаваемый адрес. Значение в "n" указывает количество следующих друг за другом адресов. Данные передаются блоками по "n" адресов в целевой адрес, начиная с *d*.



Передача данных возможна даже в том случае, если источник и цель содержат одинаковые адреса. Передача в операнды с меньшим адресом начинается с *s*, а передача в операнды с более высоким адресом – с *s + (n - 1)*.

Если *s* является словным операндом, а *d* – битовым, то в битовый операнд передаются разряды словного операнда, указанные путем обозначения битов. Например, если в *d* указан операнд K1Y30, передаются 4 младших бита словного операнда, указанного в *s*.



Если адреса представляют собой битовые операнды, количество битов в *d* и *s* должно быть одинаковым.

**ПРИМЕЧАНИЕ** Команды RBMOV и RBMOV P полезны, если при использовании процессора Q(H) требуется передать большие количества данных из регистров файлов. В случае процессоров серии "Q" эти команды работают аналогично командам BMOV. В следующих таблицах сравнивается скорость работы команд RBMOV и BMOV.

Процессор QnH				
Целевая память	Команда RBMOV		Команда BMOV	
	Время (мкс) для передачи		Время (мкс) для передачи	
	100 слов	1000 слов	100 слов	1000 слов
SRAM	56.30	367.73	44.37	393.14
Встроенная память RAM	44.37	393.14		
Флэш-ROM	29	308		

Процессор Qn				
Целевая память	Команда RBMOV		Команда BMOV	
	Время (мкс) для передачи		Время (мкс) для передачи	
	100 слов	1000 слов	100 слов	1000 слов
SRAM	115.89	579.47	63.83	535.23
Встроенная память RAM				
Флэш-ROM				

#### Источники ошибок

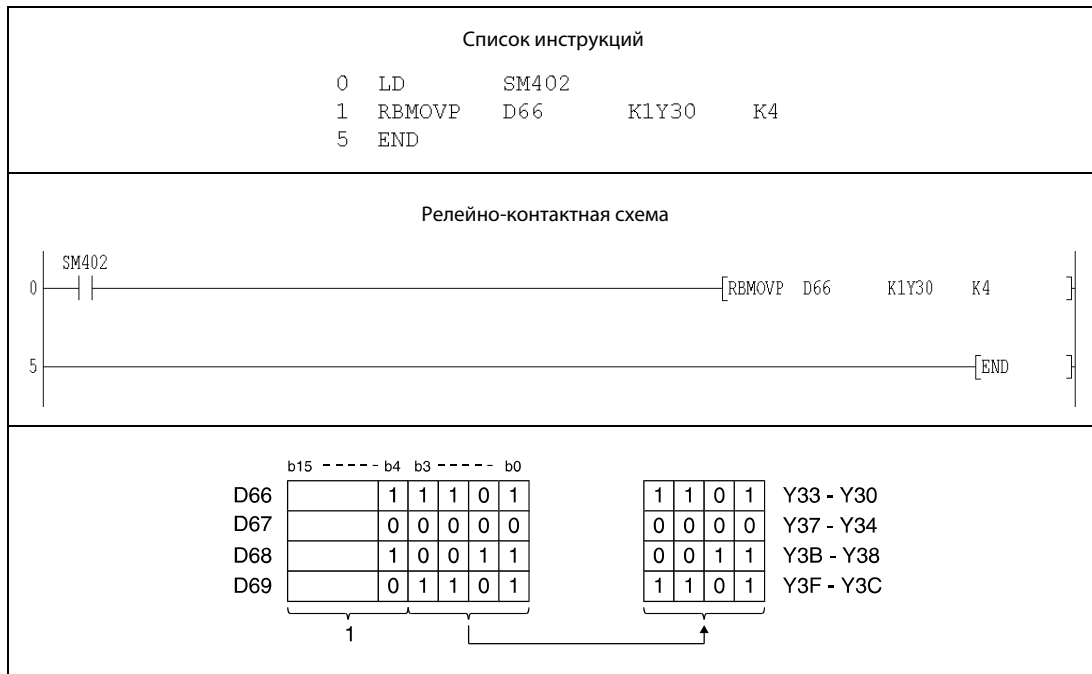
В следующих случаях возникает ошибка обработки и устанавливается флаг ошибки:

- Указанное в n количество блоков данных в s и d находится вне области операндов, предусмотренной для сохранения (код ошибки 4101).
- В s и d указан иной операнд кроме регистра файлов (код ошибки 4101).

**Пример 1** RBMOV P

Следующая программа при положительном фронте сигнала SM402 передает данные 4 младших битов (от b0 до b3) из регистров D66...D69 на выходы Y30...Y3F. Количество передаваемых блоков (4) указано константой K4.

На битовых схемах показана структура битов перед передачей и после передачи.



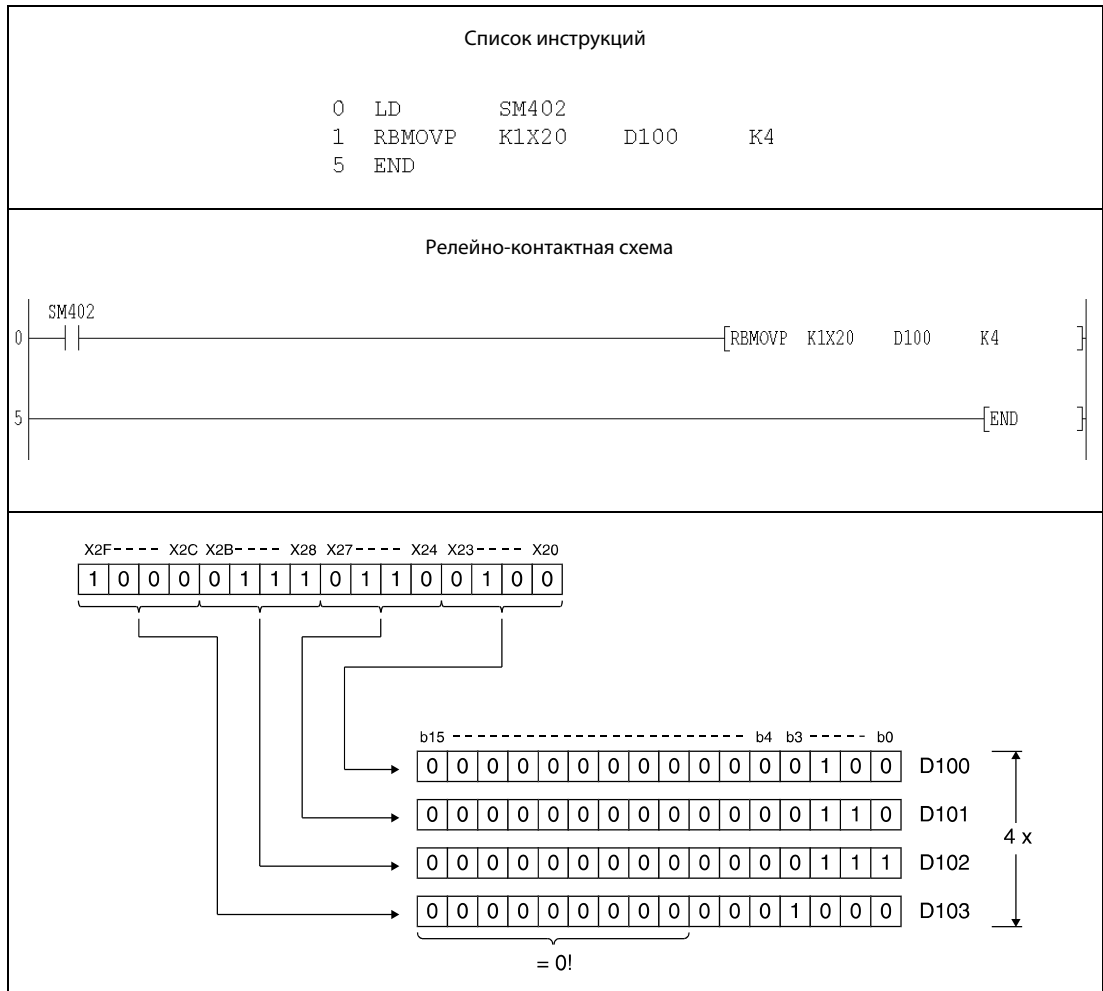
<sup>1</sup> Эти биты при операции игнорируются.



**Пример 2** RBMOV

Следующая программа при положительном фронте сигнала SM402 передает данные входов X20...X2F в регистры D100...D103. Количество блоков данных (4) указано константой K4.

На битовых схемах показана структура битов перед передачей и после передачи.



## 9.6 Команды для многопроцессорного режима

### 9.6.1 S.TO, SP.TO

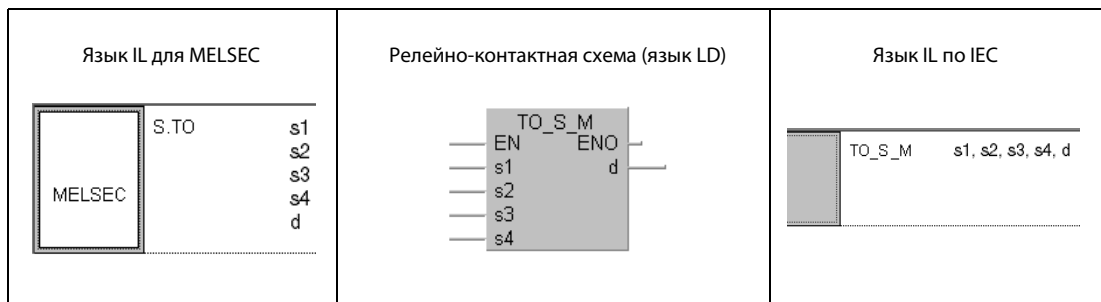
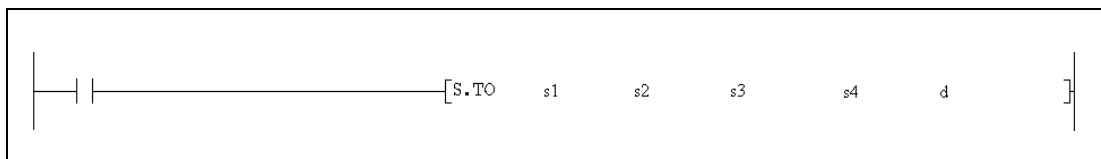
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					● <sup>1</sup>

<sup>1</sup> Только для процессоров Q02, Q02H, Q06H, Q12H и Q25H, начиная с версии "B" программного обеспечения.

Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	●	—	SM0	5
s2	—	●	●	—	—	—	—	●	—		
s3	—	●	●	—	—	—	—	—	—		
s4	—	●	●	—	—	—	—	●	—		
d	●	●	●	—	—	—	—	—	—		

GX IEC  
DeveloperGX  
Developer

Переменные

Операнд	Значение	Тип данных
s1	Головной адрес центрального процессора, в котором выполняется команда S.TO	BIN, 16 бит
s2	Начальный адрес в пользовательской области общей памяти для многопроцессорного режима	
s3	Начальный адрес области, в которой сохранены записываемые данные.	
s4	Количество записываемых слов данных (от 1 до 256)	
d	Битовый операнд, который после записи устанавливается на один программный цикл.	бит

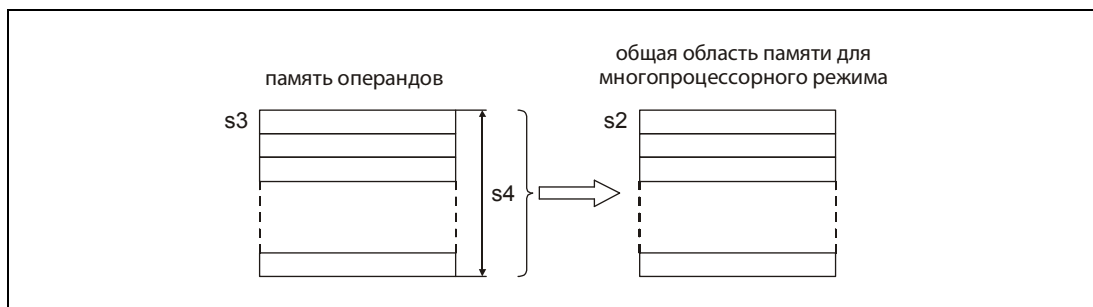
**Принцип действия**

**Ввод данных в общую область памяти для многопроцессорного режима**

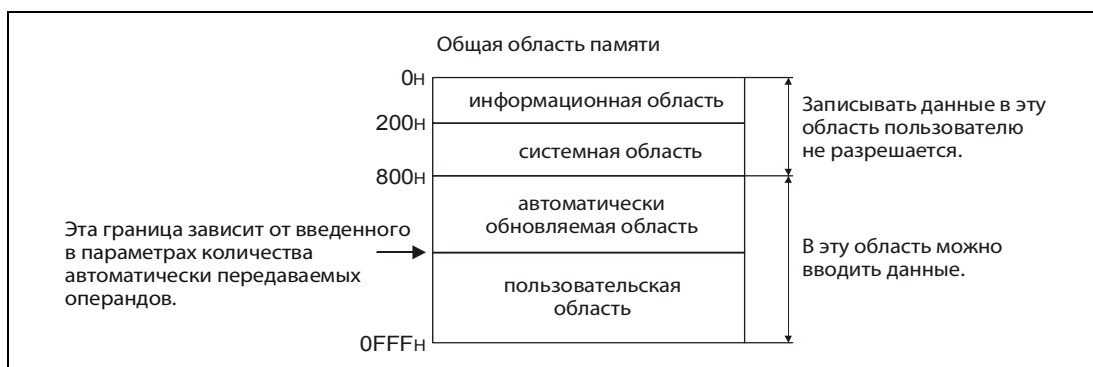
**S.TO/SP.TO**

**Команда записи**

Команда S.TO передает указанное в s4 количество слов данных в общую память центрального процессора, в котором выполняется команда S.TO. Начальный адрес исходной области в том же ЦП указывается в s3. Целевой адрес данных в общей области памяти указывается в s2. С помощью команды S.TO невозможно передавать данные непосредственно в другие процессоры многопроцессорной системы.



Общая область памяти центрального процессора "System Q" используется для обмена данными с другими модулями центральных процессоров многопроцессорной системы. С адреса 800H начинается автоматически обновляемая область, с которой граничит пользовательская область.



Головной адрес центрального процессора задан слотом центрального процессора. В s1 вводятся первые 3 знака головного адреса:

Разъем (слот) на монтажной шине	Процессор	0	1	2
Номер процессора в многопроцессорной системе	1	2	3	4
Головной адрес	3E00	3E10	3E20	3E30
Содержимое s1	3E0	3E1	3E2	3E3

Если указанное в s4 количество записываемых слов равно 0, команда не выполняется и указанный в d битовый операнд не устанавливается.

**ПРИМЕЧАНИЕ**

В каждом процессоре за один программный цикл разрешается выполнять только одну команду S.TO. Если в одном цикле выполняются несколько команд S.TO, то внутренние блокировки позволяют выполнение только команды S.TO, запущенной первой.

**Источники ошибок**

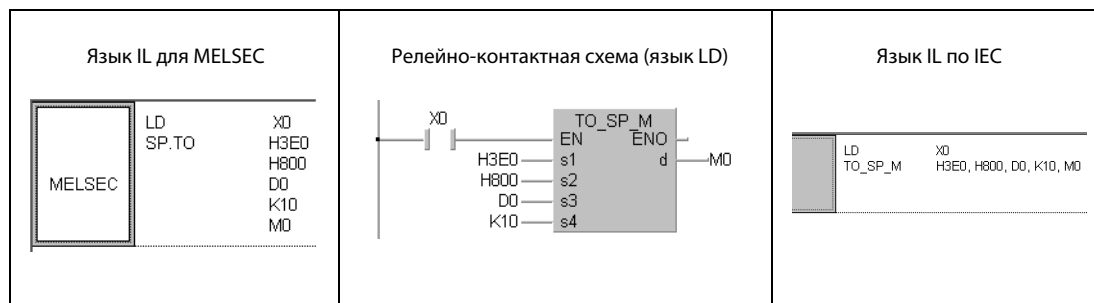
В следующих случаях возникает ошибка обработки, устанавливается флаг ошибки и в регистр SD0 записывается код ошибки:

- Указанное в s4 количество слов данных находится вне диапазона от 0 до 256 (код ошибки 4101).
- Указанный в s2 начальный адрес в пользовательской области общей памяти превышает диапазон адресов общей памяти (код ошибки 4101).
- Указанный в s2 начальный адрес с учетом указанного в s4 количества слов данных превышает диапазон адресов общей памяти (код ошибки 4101).
- Указанный в s3 начальный адрес с учетом указанного в s4 количества слов данных превышает диапазон адресов области, в которой сохраняются записываемые данные (код ошибки 4101).
- Указанный в s1 адрес не является головным адресом центрального процессора, в котором выполняется команда S.TO (код ошибки 2107).
- В s1 указан недопустимый головной адрес (3E0<sub>H</sub>, 3E1<sub>H</sub>, 3E2<sub>H</sub> или 3E3<sub>H</sub>) (код ошибки 4100).
- Указана недопустимая команда (код ошибки 4002).
- Указано неправильное количество операндов (код ошибки 4003).
- Указан недопустимый операнд (код ошибки 4002).

**Пример**

SP.TO

При включенном входе X0 содержимое регистров данных D0...D9 центрального процессора 1 записывается в пользовательскую область общей памяти этого же центрального процессора, начиная с адреса 800<sub>H</sub>.



### 9.6.2 FROM, FROMP

Процессор

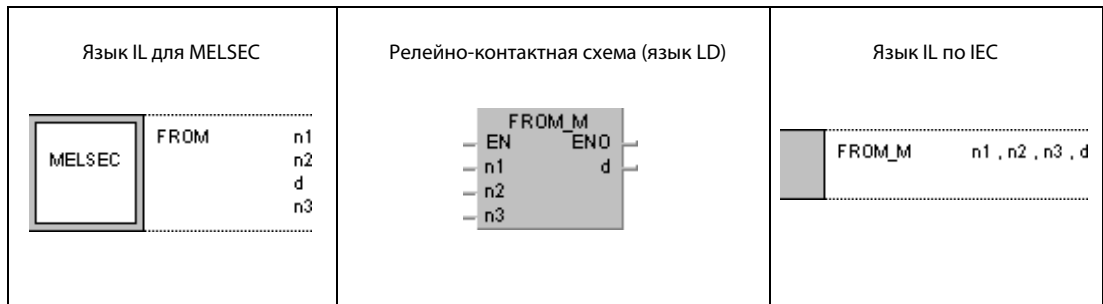
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					● <sup>1</sup>

<sup>1</sup> Только для процессоров Q02, Q02H, Q06H, Q12H и Q25H, начиная с версии "B" программного обеспечения.

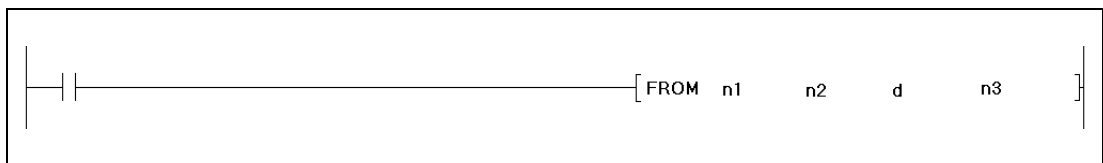
Операнды MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непос. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные U		
	битовые	словные		битовые	словные						
n1	—	●	●	●	●	●	●	●	●	SM0	5
n2	—	●	●	●	●	●	—	—			
d	—	●	●	—	—	—	—	—			
n3	—	●	●	●	●	●	—	—			

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Тип данных
n1	Головной адрес центрального процессора, из которого считываются данные.	BIN, 16 бит
n2	Начальный адрес в пользовательской области общей памяти, из которой считываются данные.	
d	Начальный адрес области, в которой сохраняются считанные данные.	
n3	Количество считываемых слов данных (от 1 до 6144)	

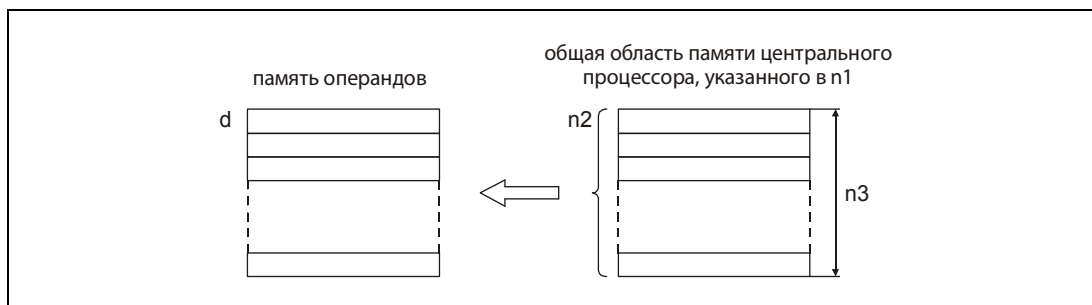
ПРИМЕЧАНИЕ

В разделе 7.8.1 описано, как можно использовать команду FROM для считывания данных из специального модуля.

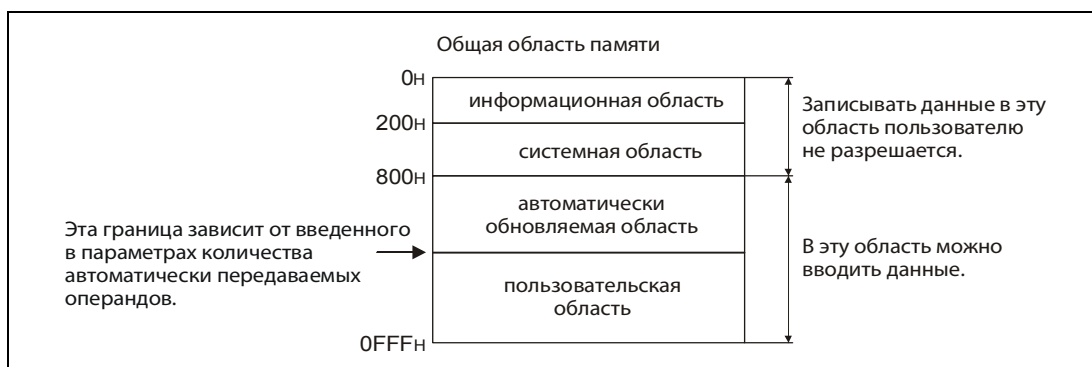
**Принцип действия**

**Считывание данных из общей области памяти другого центрального процессора FROM/FROMP Считывание словных данных**

В многопроцессорной системе команда FROM считывает словные данные из пользовательской области общей памяти другого центрального процессора. Головной адрес этого центрального процессора задается в n1. Количество передаваемых слов данных указывается в n3. Начальный адрес данных в другом центральном процессоре указывается в n2. В d указывается, где следует сохранить данные в том центральном процессоре, в котором выполняется команда FROM.



Общая область памяти центрального процессора "System Q" используется для обмена данными с другими модулями центральных процессоров многопроцессорной системы. С адреса 800H начинается автоматически обновляемая область, с которой граничит пользовательская область.



Головной адрес центрального процессора задан слотом центрального процессора. В n1 вводятся первые 3 знака головного адреса:

Разъем (слот) на монтажной шине	Процессор	0	1	2
Номер процессора в многопроцессорной системе	1	2	3	4
Головной адрес	3E00	3E10	3E20	3E30
Содержимое n1	3E0	3E1	3E2	3E3

После считывания данных устанавливается специальный маркер SM390. Если центральный процессор, из которого требуется считать данные, находится в состоянии сброса, маркер SM390 не устанавливается. В этом случае сообщение об ошибке не выводится.

Если указанное в n3 количество считываемых слов равно 0, команда не выполняется.

**Источники ошибок**

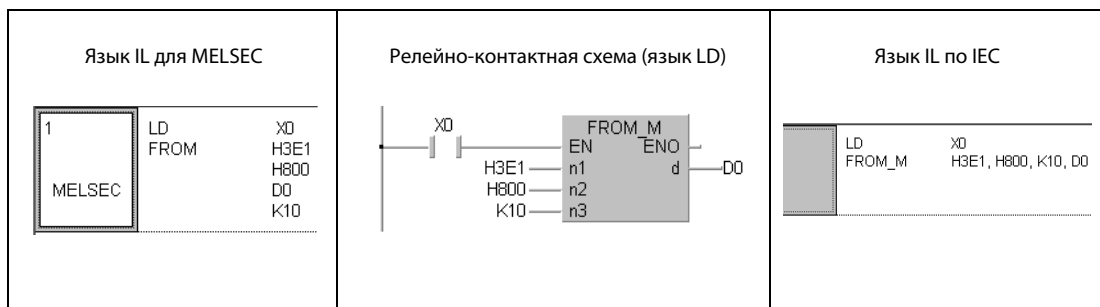
В следующих случаях возникает ошибка обработки, устанавливается флаг ошибки и в регистр SD0 записывается код ошибки:

- Указанный в n2 начальный адрес пользовательской области превышает диапазон адресов общей памяти (код ошибки 4101).
- Указанный в n2 начальный адрес с учетом указанного в n3 количества слов данных превышает диапазон адресов общей памяти (код ошибки 4101).
- Указанный в d начальный адрес с учетом указанного в n3 количества слов данных превышает диапазон адресов области, в которой сохраняются считанные данные (код ошибки 4101).
- В n1 указан головной адрес центрального процессора, в котором выполняется команда FROM (код ошибки 2114).
- В слоте, указанном с помощью головного адреса в n1, нет модуля центрального процессора (код ошибки 2110).

**Пример**

FROM

При включенном входе X0 из пользовательской области общей памяти в центральном процессоре 2, начиная с адреса 800<sub>H</sub>, считываются десять слов данных. Эти данные записываются в регистры данных D0...D9 того центрального процессора, в котором выполняется команда FROM.







## 10 Команды для Q4ARCPU

Имеется возможность построить избыточную систему управления на основе двух модулей центральных процессоров типа Q4ARCPU. При выходе из строя одного центрального процессора второй процессор безразрывно перенимает задачи управления на себя и обеспечивает продолжение процесса. Типичными областями применения избыточных систем являются, например, электростанции и технологические процессы в химической промышленности или на водопроводных станциях.

Описываемые в этом разделе команды можно использовать только для процессоров Q4ARCPU.

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту IEC
Режим при запуске центрального процессора	S.STMODE	STMODE_S_M
Режим при переключении центрального процессора	S.CGMODE	CGMODE_S_M
Передача данных для резервного центрального процессора	S.TRUCK	TRUCK_S_M
Ввод данных в буферную память специальных модулей или считывание данных из буферной памяти	S.SPREF	SPRE_S_M

# 10.1 Команды выбора режима

## 10.1.1 SТMODE

Процессор

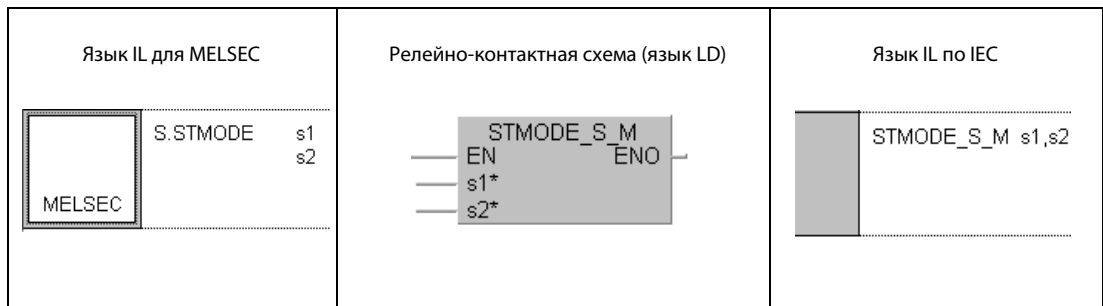
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				● <sup>1</sup>	

<sup>1</sup> только для Q4AR

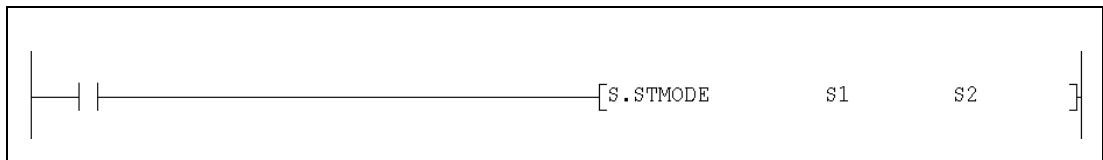
Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	—	—	—	—	—	—	●	—	—	
s2	—	—	—	—	—	—	—	●	—		

GX IEC  
Developer



GX  
Developer



Переменные

Операнд	Значение	Тип данных
s1	Выбор режима при запуске центрального процессора (0 = новый запуск, 1 = горячий перезапуск)	BIN, 16 бит
s2	Максимальное время исчезновения напряжения, после которого происходит перезапуск.	BIN, 16 бит

**Принцип действия**

**Выбор режима работы центрального процессора при включении напряжения питания**  
**STMODE**                      **Выбор режима при запуске**

С помощью s1 можно выбрать, должны ли при включении напряжения питания и последующем запуске центрального процессора стираться операнды центрального процессора (новый запуск), или эти операнды должны сохранять свои значения (горячий перезапуск).

Если напряжение исчезло на время, превышающее настройку s2, то даже при выбранном горячем перезапуске данные автоматически стираются и происходит новый запуск.

Эта команда выполняется при включении напряжения – даже в том случае, если условие выполнения команды не включено. Условие выполнения рассматривается в качестве фиктивного контакта. При включении условия выполнения в нормальном программном цикле выполняется команда NOP.

В каждой системе эту команду требуется выполнить только один раз. Если в системе имеется несколько программ, эта команда должна встречаться только один раз. Если эта команда запрограммирована многократно, правильное выполнение команды не гарантировано.

Содержимое s1 может иметь значение только 0 или 1:

0: новый запуск (операнды вне области фиксации стираются)

1: горячий перезапуск (операнды не стираются как при новом запуске, однако индексные регистры и результаты (поток сигналов) стираются, а специальные маркеры и регистры устанавливаются на значения предварительной настройки)

Время в s2 указывается в секундах (от 0 до 65535). Если установлено время 0, новый запуск не выполняется. Значения, превышающие 32767, должны указываться в шестнадцатеричном виде.

**Источники ошибок**

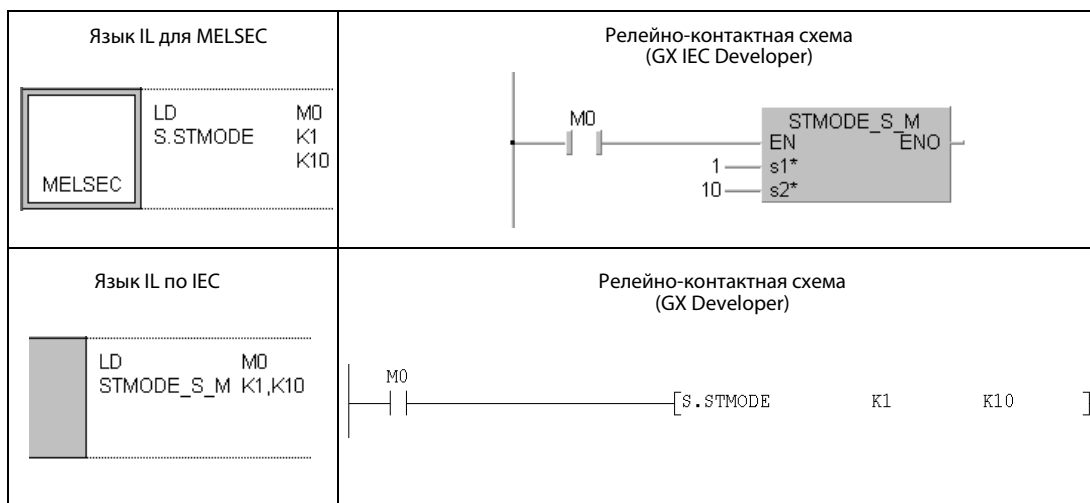
В следующем случае возникает ошибка обработки, устанавливается флаг ошибки (SM0) и в специальный регистр SD0 записывается код ошибки:

- В s1 или s2 указаны значения вне допустимого диапазона (код ошибки: 4104).

**Пример**

STMODE

В следующем фрагменте программы выбирается горячий перезапуск. Если напряжение исчезло дольше чем на 10 секунд, то при его появлении происходит новый запуск.



### 10.1.2 CGMODE

**Процессор**

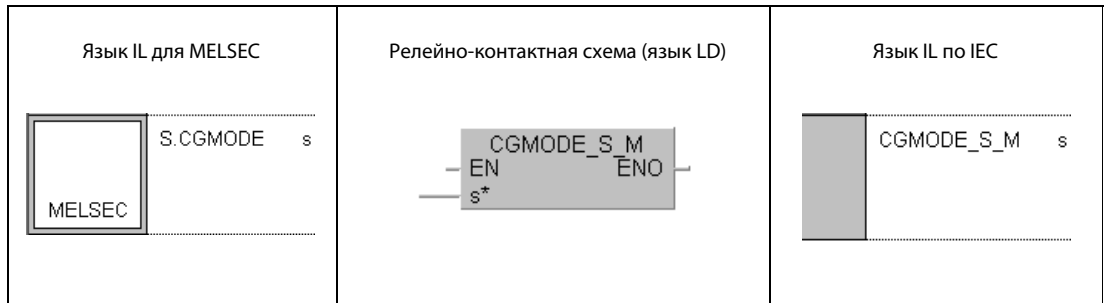
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				● <sup>1</sup>	

<sup>1</sup> только для Q4AR

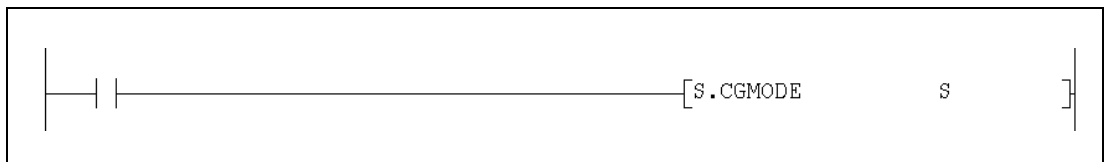
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непос. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	—	—	—	—	—	—	—	●	—	—	

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
s	Выбор режима	BIN, 16 бит

**Принцип действия** **Выбор режима при переключении модулей центральных процессоров**  
**CGMODE** **Выбор режима**

С помощью этой команды можно выбрать, должны ли при переключении на резервную систему стираться операнды центрального процессора, или они должны сохранять свои значения. Выбор определяется содержимым переменной s.

Эта команда выполняется при включении напряжения – даже в том случае, если условие выполнения команды не включено. Условие выполнения рассматривается в качестве фиктивного контакта. При включении условия выполнения в нормальном программном цикле выполняется команда NOP.

В каждой системе эту команду можно запрограммировать только один раз. Если имеется несколько программ, используйте эту команду только в одной программе. Если эта команда запрограммирована многократно, правильное выполнение команды не гарантировано.

Содержимое s может иметь значение только 0 или 1:

0: новый запуск (операнды вне области фиксации стираются)

1: горячий перезапуск (операнды и результаты не стираются, как при новом запуске, однако специальные маркеры и регистры устанавливаются на значения предварительной настройки).

**Источники ошибок**

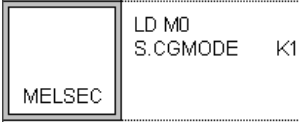
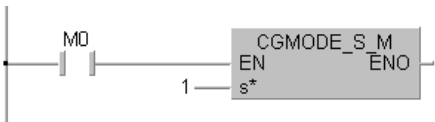
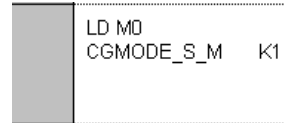
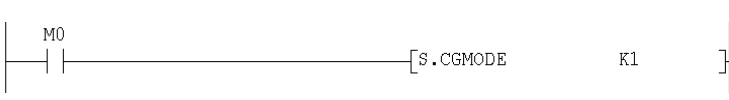
В следующем случае возникает ошибка обработки, устанавливается флаг ошибки (SM0) и в специальный регистр SD0 записывается код ошибки:

- В s указано значение вне допустимого диапазона (код ошибки: 4104).

**Пример**

CGMODE

В следующем фрагменте программы выбирается горячий перезапуск центрального процессора при переключении на резервную систему.

<p>Язык IL для MELSEC</p>  <pre>LD M0 S.CGMODE K1</pre>	<p>Релейно-контактная схема (GX IEC Developer)</p> 
<p>Язык IL по IEC</p>  <pre>LD M0 CGMODE_S_M K1</pre>	<p>Релейно-контактная схема (GX Developer)</p> 

## 10.2 Команды передачи данных

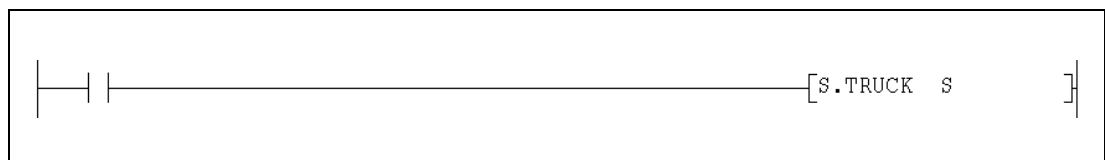
### 10.2.1 TRUCK

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				● <sup>1</sup>	

<sup>1</sup> только для Q4ARОперанды  
MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s	—	● <sup>1</sup>	●	—	—	—	—	—	—	—	—

<sup>1</sup> Допускаются только операнды из области фиксации.GX IEC  
DeveloperGX  
Developer

Переменные

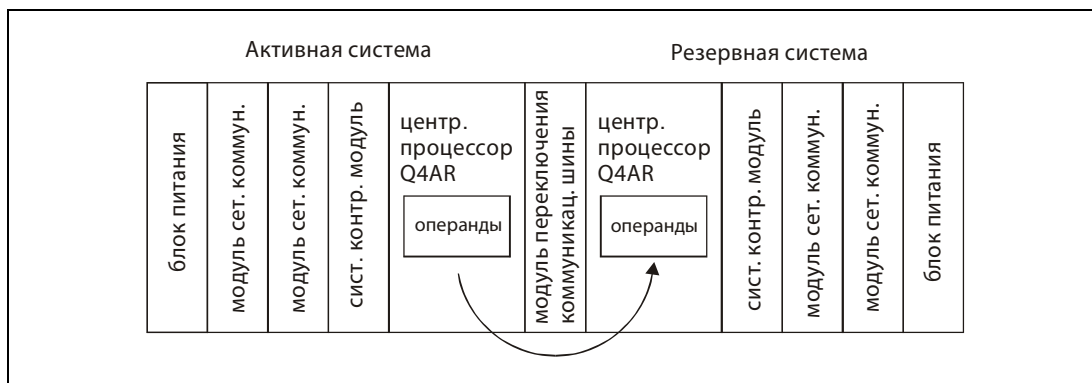
Операнд	Значение	Тип данных
s	Начальный адрес блока параметров	BIN, 16 бит

**Принцип действия**

**Обмен данными между модулями центральных процессоров избыточной системы TRUCK Передача данных в резервный центральный процессор**

В избыточной системе в конце программного цикла (при обработке команды END) данные из активного центрального процессора Q4AR передаются в резервный центральный процессор (которым также является процессор Q4AR).

Команда TRUCK устанавливает, какие данные должны передаваться. В переменной *s* указывается начальный адрес блока параметров, в котором, в свою очередь, указан тип и количество передаваемых операндов.



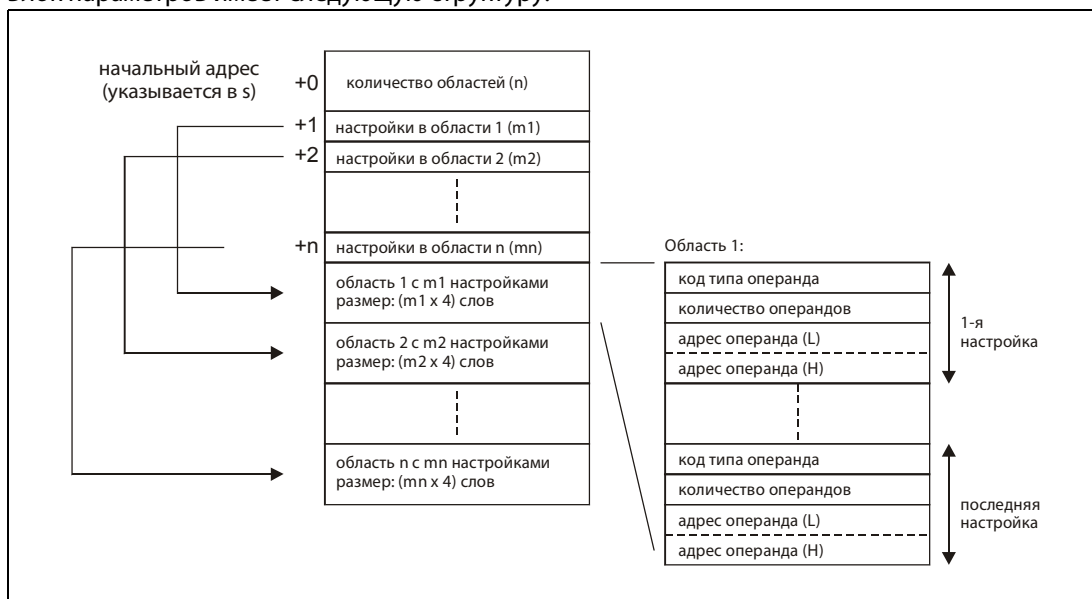
В каждой системе эту команду можно запрограммировать только один раз. Если имеется несколько программ, используйте эту команду только в одной программе. Если эта команда запрограммирована многократно, правильное выполнение команды не гарантировано.

Центральный процессор Q4AR считывает содержимое блока параметров при включении напряжения питания или при сбросе. (Учитывайте, что если содержимое блока параметров было изменено, систему необходимо перезапустить.) Блок параметров разделен на несколько областей. Передача областей управляется специальными маркерами с SM1520 по SM1583. По сигналу SM1520 передается первая область, по сигналу SM1521 – вторая и так далее. Команда S.TRUCK должна выполняться после установки соответствующих специальных маркеров.

**ПРИМЕЧАНИЕ**

Для передачи данных с помощью команды S.TRUCK можно использовать те же специальные маркеры (с SM1520 по SM1583), что и для обновления буферной памяти с помощью команды S.PREF.

Блок параметров имеет следующую структуру:



**Содержимое блока параметров:**

- Количество областей (n)  
В блоке параметров может иметься несколько областей, в которых, например, указаны начальные адреса и количество передаваемых операндов. Введите здесь количество областей.
- Количество настроек на каждую область (от m1 до mn)  
Каждая область может содержать несколько настроек передачи данных. Каждая настройка состоит из типа операнда, количества операндов и начального адреса.
- Области с настройками  
Каждая настройка для обмена данными занимает 4 слова:  
1-е слово: кодированное указание типа операнда (см. следующую таблицу)

Операнд	Код	Операнд	Код	Операнд	Код	Операнд	Код
X	0	B	5	C <sup>1</sup>	10	Z	15
Y	1	F	6	D	11	SB	16
M	2	V	7	W	12	SW	17
L	3	ST	8	R	13	SM	18
S	4	T <sup>1</sup>	9	ZR	14	SD	19

<sup>1</sup> В случае таймеров (T) и счетчиков (C) сюда входит контакт, катушка и текущее значение.

**ПРИМЕЧАНИЕ** Операнды, объявленные в качестве локальных, не передаются.

2-е слово: количество операндов

Количество операндов можно указать в десятичном или шестнадцатеричном виде. В случае битовых операндов указанное количество должно быть кратно 16.

3-е и 4-е слово: начальный адрес, младший (L) и старший байт (H)

Начальный адрес можно указать в десятичном или шестнадцатеричном виде. В случае битовых операндов указанный адрес может быть либо 0, или представлять собой число, кратное 16 (0, 16, 32, ...).

**ПРИМЕЧАНИЯ** Учитывайте следующие ограничения:

- В пределах одного блока параметров можно указать максимум 64 области ( $n \leq 64$ ).
- Сумма настроек не должна превышать 2048 ( $m1 + m2 + \dots + mn \leq 2048$ ).
- Если для количества настроек области (от m1 до mn) указан 0, количество областей также устанавливается на 0. В результате эта область пропускается.
- В конце одного программного цикла можно передать максимум 48 килослов. При превышении этого количества данных возникает ошибка и передача данных не происходит.
- При указании битовых операндов количество операндов и их начальный адрес должны быть кратны 16.
- При указании таймеров или счетчиков действительно переданное количество операндов определяется по следующей формуле:

$$\text{Переданное количество операндов} = \text{настроенное количество операндов} \times (1 + 1/8)$$

"1" в скобках означает фактическое значение таймера/счетчика

(так как оно отображается одним словом), а значение "1/8" – бит контакта или катушки.



**Управление выполнением команды TRUCK**

На выполнение команды TRUCK можно влиять с помощью специального маркера SM1518. Сделанный выбор начинает действовать после обработки команды END того программного цикла, в котором был установлен или сброшен маркер SM1518.

**а) Передача со временем ожидания (SM1518 = 0)**

Если в этот момент резервная система обращается к области памяти, в которую требуется передать данные, активный центральный процессор ждет, пока резервная система не завершит свою операцию. Это время ожидания активного центрального процессора удлинит время цикла.

**б) Повторение передачи (SM1518 = 1)**

Если в этот момент резервная система обращается к области памяти, в которую требуется передать данные, активный центральный процессор выполняет очередную команду END без передачи данных. Во время повторения передачи прочие запросы на передачу данных игнорируются. Время цикла активного центрального процессора не удлиняется.

**Индикация выполнения команды TRUCK**

После передачи на время одного цикла устанавливается по одному специальному маркеру (от SM1712 до SM1775) для каждой переданной области. (Область 1: SM1712, область 2: SM1713 ... область 64: SM1775)

**Источники ошибок**

В следующих случаях возникает ошибка обработки, устанавливается флаг ошибки SM0 и в специальный регистр SD0 записывается код ошибки:

- Файл с регистром файлов не существует, хотя в блоке параметров указан регистр файлов R (код ошибки: 2402).
- Указано значение вне допустимого диапазона (код ошибки: 4104).
- Количество передаваемых данных превышает 48 килослов (код ошибки: 4104).

**Пример**

**TRUCK**

В резервный центральный процессор передаются состояния маркеров M0...M95 и M320...M639, а также содержимое регистров данных D0...D29 и D600...D699. Блок параметров начинается с R100 и содержит две области, управляемых с помощью специальных маркеров SM1520 и SM1521. В первой области указаны передаваемые маркеры, а во второй области - передаваемые регистры данных:

Блок параметров		Описание	Примечание	
Адрес	Содержимое			
R100	2	Количество областей	—	
R101	2	Настройки в области 1	—	
R102	2	Настройки в области 2	—	
R103	2	Код операнда (2 = M)	область 1	
R104	96	Количество операндов		
R105	0	Начальный адрес операндов (M0)		
R106	0			
R107	2	Код операнда (2 = M)		
R108	320	Количество операндов		настройка 2
R109	320	Начальный адрес операндов (M320)		
R110	0		область 2	
R111	11	Код операнда (11 = R)		
R112	30	Количество операндов		
R113	0	Начальный адрес операндов (D0)		
R114	0			
R115	11	Код операнда (11 = R)		
R116	100	Количество операндов		настройка 2
R117	600	Начальный адрес операндов (D600)		
R118	0			

<p>Язык IL для MELSEC</p> <pre> MELSEC LD      M0 OUT     SM1520 OUT     SM1521  MELSEC LD      M10 S.TRUCK R100         </pre>	<p>Релейно-контактная схема (GX IEC Developer)</p>
<p>Язык IL по IEC</p> <pre> LD      M0 ST      SM1520 ST      SM1521  LD      M10 TRUCK_S_M R100         </pre>	<p>Релейно-контактная схема (GX Developer)</p>

## 10.2.2 SPREF

### Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				● <sup>1</sup>	

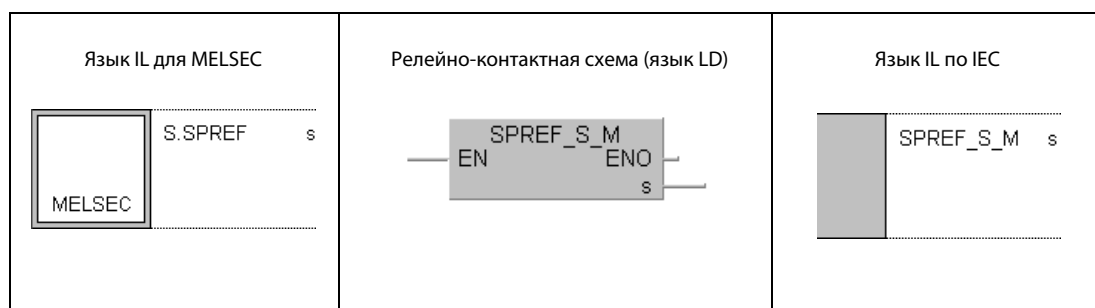
<sup>1</sup> только для Q4AR

### Операнды MELSEC Q

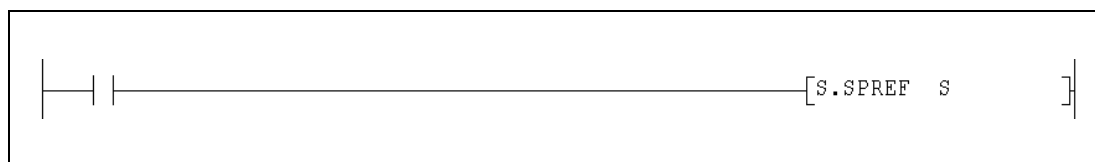
s	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
—	● <sup>1</sup>	●	—	—	—	—	—	—	—	—	

<sup>1</sup> Допускаются только операнды из области фиксации.

### GX IEC Developer



### GX Developer



### Переменные

Операнд	Значение	Тип данных
s	Начальный адрес блока параметров	BIN, 16 бит

### Принцип действия

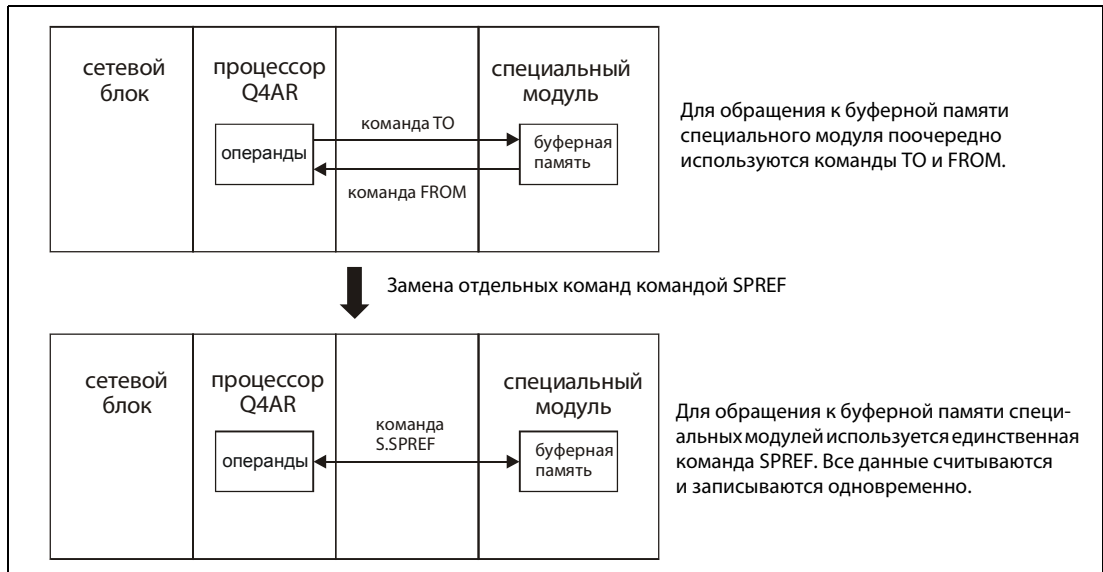
**Обмен данными между центральным процессором и буферной памятью специальных модулей**

#### S.SPREF Обмен данными с буферной памятью

Команда SPREF позволяет осуществлять обмен данными между центральным процессором Q4AR и буферной памятью одного или нескольких специальных модулей.

### ПРИМЕЧАНИЕ

Команда S.PREF не может использоваться для специальных модулей в децентрализованных станциях ввода-вывода MELSECNET (II), /B или /10, а также в MELSECNET/MINI-S3.

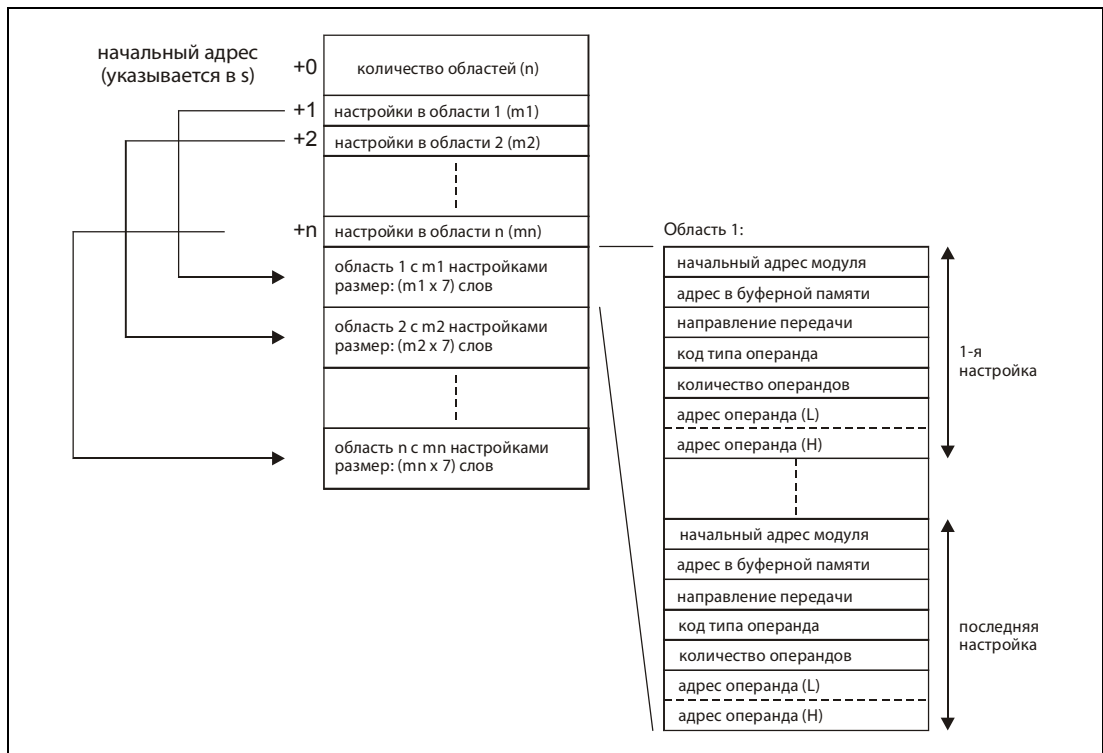


В переменной *s* указывается начальный адрес блока параметров, содержащего настройки для обмена данными. Содержимое блока параметров должно быть установлено перед выполнением команды SPREF. Блок параметров разделен на несколько областей. Передача областей управляется специальными маркерами с SM1520 по SM1583. По сигналу SM1520 передается первая область, по сигналу SM1521 – вторая и так далее. Соответствующие специальные маркеры должны быть установлены перед выполнением команды SPREF.

**ПРИМЕЧАНИЕ**

Для передачи данных с помощью команды SPREF можно использовать те же специальные маркеры (с SM1520 по SM1583), что и для команды TRUCK.

Блок параметров имеет следующую структуру:



**Содержимое блока параметров:**

- Количество областей (n)  
В блоке параметров может иметься несколько областей, в которых, например, указаны начальный адрес специального модуля и количество передаваемых операндов. Введите здесь количество таких областей.
- Количество настроек на каждую область (от m1 до mn)  
Каждая область может содержать несколько настроек для обмена данными. Каждая настройка состоит из нижеприведенных элементов.
- Области с настройками  
Каждая настройка занимает 7 слов области и содержит параметры для передачи данных.
  - 1-е слово: головной адрес специального модуля на монтажной шине  
Указываются только два первых разряда 3-разрядного адреса (пример: начальный адрес X/Y100 указывается в виде 10H).
  - 2-е слово: адрес в буферной памяти  
Адрес в буферной памяти можно указывать в десятичном или шестнадцатеричном виде.
  - 3-е слово: направление потока данных  
Выбор считывания данных из специального модуля или записи в специальный модуль.  
0 = считывание (из буферной памяти в центральный процессор), 1 = запись (из центрального процессора в специальный модуль)
  - 4-е слово: кодированное указание типа операнда (см. следующую таблицу)

Операнд	Код	Операнд	Код	Операнд	Код	Операнд	Код
X	0	B	5	C <sup>1</sup>	10	Z	15
Y	1	F	6	D	11	SB	16
M	2	-	-	W	12	SW	17
L	3	ST	8	R	13	SM	18
—	—	T <sup>1</sup>	9	ZR	14	SD	19

<sup>1</sup> В случае таймеров (T) и счетчиков (C) передается только их фактическое значение.

- 5-е слово: количество операндов  
Количество операндов можно указать в десятичном или шестнадцатеричном виде. В случае битовых операндов указанное количество должно быть кратно 16.
- 6-е и 7-е слово: начальный адрес операндов, младший (L), старший байт (H)  
Начальный адрес операндов занимает два слова. Его можно указывать в десятичном или шестнадцатеричном виде. В случае битовых операндов указанный адрес может быть либо 0, или представлять собой число, кратное 16 (0, 16, 32, ...).

**ПРИМЕНЕНИЯ** *Учитывайте следующие ограничения:*

- В пределах одного блока параметров можно указать максимум 64 области ( $n \leq 64$ ).
- Сумма настроек не должна превышать 2048 ( $m1 + m2 + \dots + mn \leq 2048$ ).
- Если для количества настроек области (от m1 до mn) указан 0, то количество областей также устанавливается на 0. В результате эта область пропускается.

**Источники ошибок**

В следующем случае возникает ошибка обработки, устанавливается флаг ошибки SMO и в специальный регистр SD0 записывается код ошибки:

- Указано значение вне допустимого диапазона (код ошибки: 4104).

**Пример****SPREF**

В следующем примере осуществляется доступ к двум специальным модулям. Для каждого специального модуля в блоке параметров, сохраненном начиная с регистра файлов R100, создана отдельная область:

- 1-я область: обращение к специальному модулю с головным адресом X/Y20

Эта область передается при установленном маркере SM1520.

Содержимое адресов 0...3 буферной памяти считывается из специального модуля и передается (записывается) в регистры файлов R0...R3.

Содержимое регистров файлов R10 и R11 передается (записывается) в адреса буферной памяти 10 и 11 специального модуля.

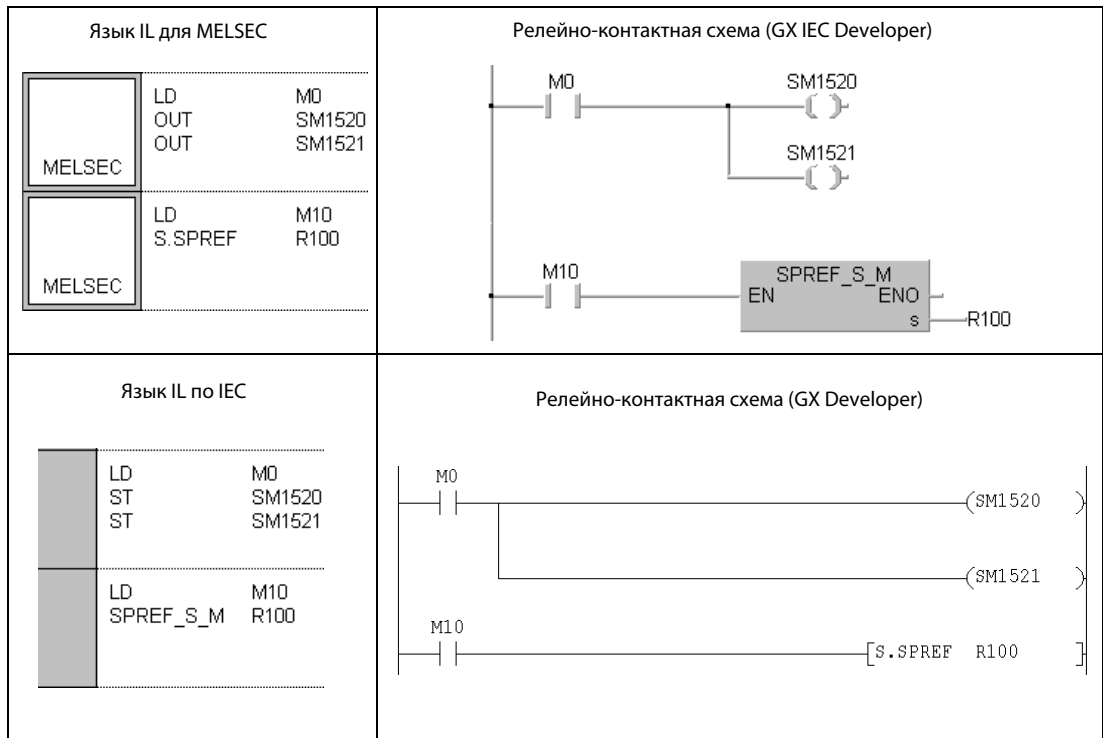
- 2-я область: обращение к специальному модулю с головным адресом X/Y100

Эта область передается при установленном маркере SM1521.

Содержимое адресов 110...119 буферной памяти считывается из специального модуля и передается (записывается) в регистры данных D110...D113.

Блок параметров для этого примера программы содержит следующие константы:

Блок параметров		Описание	Примечание		
Адрес	Содержимое				
R100	2	Количество областей	—		
R101	2	Настройки в области 1	—		
R102	1	Настройки в области 2	—		
R103	2	Головной адрес специального модуля (X/Y20)	область 1	область 1	
R104	0	Начальный адрес в буферной памяти			
R105	0	Направление потока данных (0 = считывание)			
R106	13	Код операнда (13 = R)			
R107	4	Количество операндов			
R108	0	Начальный адрес операндов (R0)			
R109	0				
R110	2	Головной адрес специального модуля (X/Y20)			настройка 2
R111	10	Начальный адрес в буферной памяти			
R112	1	Направление потока данных (1 = запись)			
R113	13	Код операнда (13 = R)			
R114	2	Количество операндов			
R115	10	Начальный адрес операндов (R10)			
R116	0				
R117	10	Головной адрес специального модуля (X/Y100)	область 2	настройка 1	
R118	110	Начальный адрес в буферной памяти			
R119	0	Направление потока данных (0 = считывание)			
R120	11	Код операнда (11 = D)			
R121	10	Количество операндов			
R122	110	Начальный адрес операндов (D110)			
R123	0				







Группа команд	Описание
Команды для модулей последовательного интерфейса	Считывание принятых данных в программе прерывания и их запись в центральный процессор контроллера, считывание, установление и стирание определенных пользователем кадров данных, передача данных на основе определенных пользователем кадров данных
Команды для модулей Profibus/DP	Считывание данных из буферной памяти модулей Profibus/DP и их запись в буферную память
Команды для модулей Ethernet	Запись данных в фиксированные буферы и их считывание из фиксированных буферов, открытие и закрытие сеанса связи, считывание и стирание буфера сообщений об ошибках, повторная инициализация модуля Ethernet
Команды для MELSECNET/10	Установление станций для попарной работы (дуплексный режим)
Команды для CC-Link	Передача параметров сети, настройка параметров автоматического обновления, считывание данных из буферной памяти другой станции CC-Link или центрального процессора контроллера этой станции, запись данных в буферную память другой станции CC-Link или центрального процессора контроллера этой станции, запись данных в автоматически обновляемую память и считывание данных из этой памяти

## 11.1 Команды для модулей последовательного интерфейса

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту IEC
Считывание принятых данных из модуля QJ71C24 в центральный процессор контроллера в программе прерывания	Z.BUFRCV5	BUFRCV5_M
Считывание определенного пользователем кадра данных из модуля интерфейса	G.GETE	GETE_M
	GP.GETE	GETEP_M
Запись или стирание определенного пользователем кадра данных в модуль интерфейса	G.PUTE	PUTE_M
	GP.PUTE	PUTEP_M
Передача данных на основе определенных пользователем кадров данных	G.PRR	PRR_M
	GP.PRR	PRRP_M

### 11.1.1 BUFRCVS

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

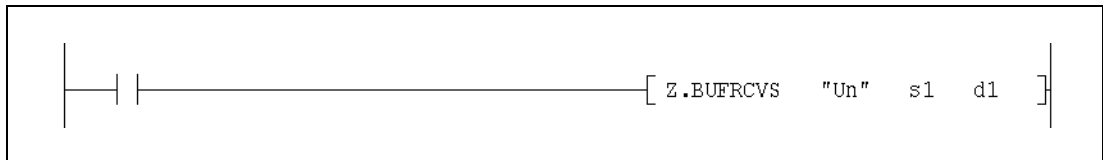
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	SM0		
d1	●	●	●	—	—	—	—	—			

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Диапазон	Устанавливает	Тип данных
„Un“	Головной адрес модуля интерфейса на монтажной шине (указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде "U10".)	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит
s1	Интерфейс, через который принимаются данные 1: интерфейс 1 (CH1) 2: интерфейс 2 (CH2)	1 или 2		
d1	Первый операнд области, в которой сохраняются принятые данные.			
	Операнд	Значение	Описание	Диапазон
	(d1) + 0	Длина данных	Количество принятых данных. Единица (байты или слова) зависит от параметрирования.	—
	от (d1) + 1 до (d1) + n	Принятые данные	В эту область поочередно, в возрастающей последовательности записываются данные, считанные из приемной области буферной памяти.	система

**Принцип действия**

**Считывание принятых данных из модуля QJ71C24**

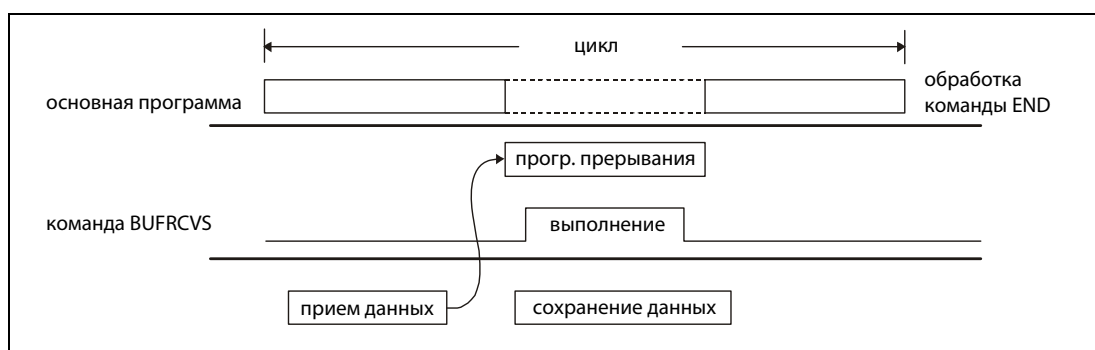
**BUFRCVS Считывание данных**

Команды BUFRCVS считывают из модуля интерфейса QJ71C24 данные, переданные в этот модуль из внешнего устройства, и сохраняют их в центральном процессоре контроллера.

При этом команда BUFRCVS самостоятельно определяет, где хранятся эти данные в буферной памяти модуля QJ71C24, и передает их в область операндов, указанную с помощью d1.

После передачи данных автоматически сбрасывается сигнал запроса на считывание принятых данных (X3/XA) или входной сигнал "Ошибка при приеме данных" (X4/XB). Устанавливать выходной сигнал (Y1/Y8), сообщающий модулю интерфейса о том, что данные считаны, в случае применения команды BUFRCVS не требуется.

Команда BUFRCVS используется в программе прерывания и завершается в том же цикле. На рисунке ниже показана диаграмма изменения сигналов при выполнении команды BUFRCVS:



**ПРИМЕЧАНИЯ**

Если принятые данные считываются командой BUFRCVS в программе прерывания, то данные того же интерфейса более не могут считываться в основной программе. Поэтому команда BUFRCVS не может использоваться в сочетании со следующими командами:

- команда INPUT
- команда BIDIN
- команда FROM в сочетании с входными и выходными сигналами модуля интерфейса

Команды BUFRCVS и CSET не могут выполняться одновременно.

Указанная в d1 область операндов в центральном процессоре контроллера, в которой сохраняются данные, должна иметь достаточный размер, чтобы вместить все данные, передаваемые внешним устройством. Если область слишком мала, то не уместившиеся данные утрачиваются.

**Источники ошибок**

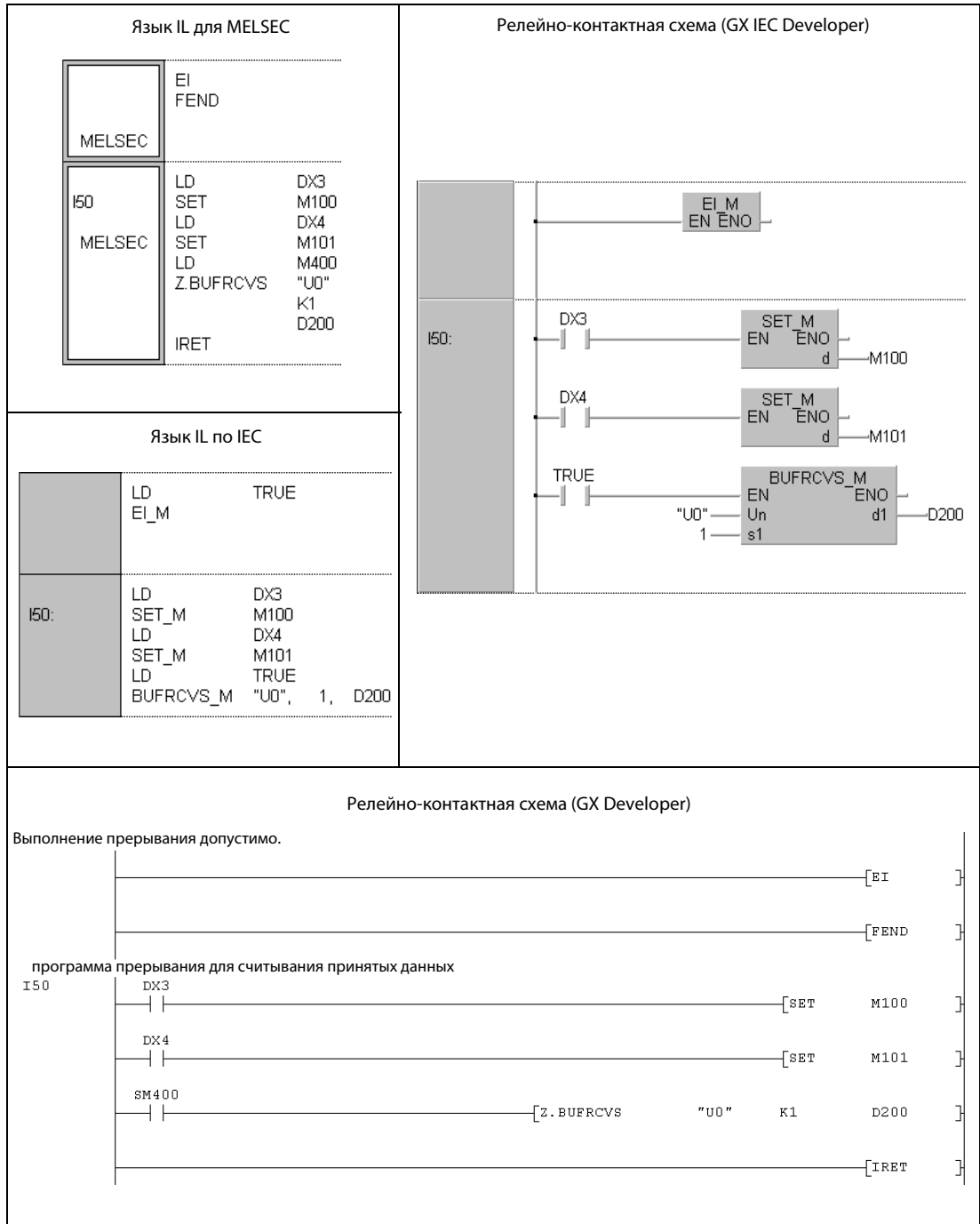
Если при выполнении команды BUFRCVS возникла ошибка, то устанавливается флаг ошибки SM0 и в регистр SD0 записывается код ошибки. Коды ошибок подробно разъяснены в следующих руководствах:

- При коде ошибки до 4FFF<sub>H</sub> указания по устранению ошибки имеются в этом руководстве (глава 13).
- При коде ошибки, начиная с 7000<sub>H</sub>, более подробную информацию можно найти в руководстве по модулю интерфейса QJ71C24.

Если данные приняты с ошибками (на что указывают входные сигналы X4 или XB), то оценивается код ошибки, который в этом случае записывается в адреса буферной памяти 258<sub>H</sub> и 268<sub>H</sub> модуля интерфейса.

**Пример** BUFRCVS

Следующая программа считывает данные, принятые через интерфейс 1 QJ71C24 (головной адрес X/Y0), и сохраняет их в регистре D200. Модуль запараметрирован так, что прерывание вызывает только интерфейс 1. При приеме данных обрабатывается программа прерывания 50 (I50). Для связи с основной программой служат маркеры M100 и M101. Если данные приняты без ошибок, в основной программе устанавливается M100, а если данные приняты с ошибками, устанавливается M101. Оба маркера сбрасываются в основной программе.



11.1.2 GETE, GETEP

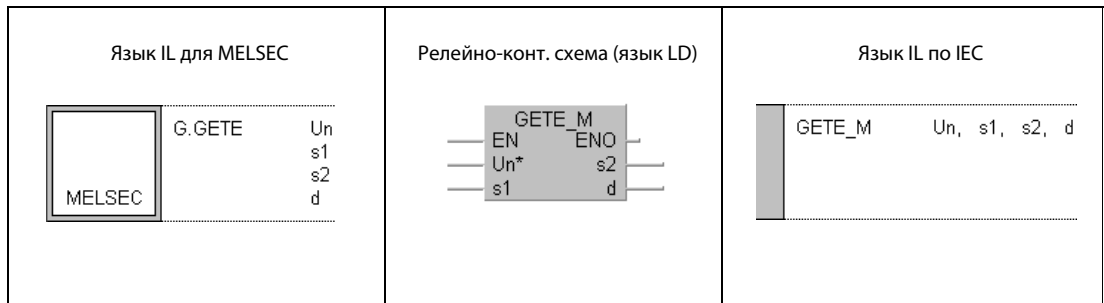
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

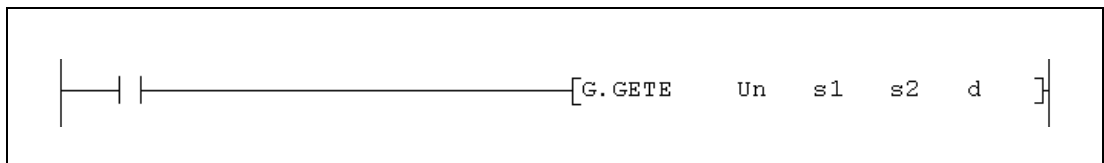
Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.ов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	—	SMO	
s2	—	●	●	—	—	—	—	—	—		
d	●	●	●	—	—	—	—	—	—		

GX IEC  
Developer



GX  
Developer



## Переменные

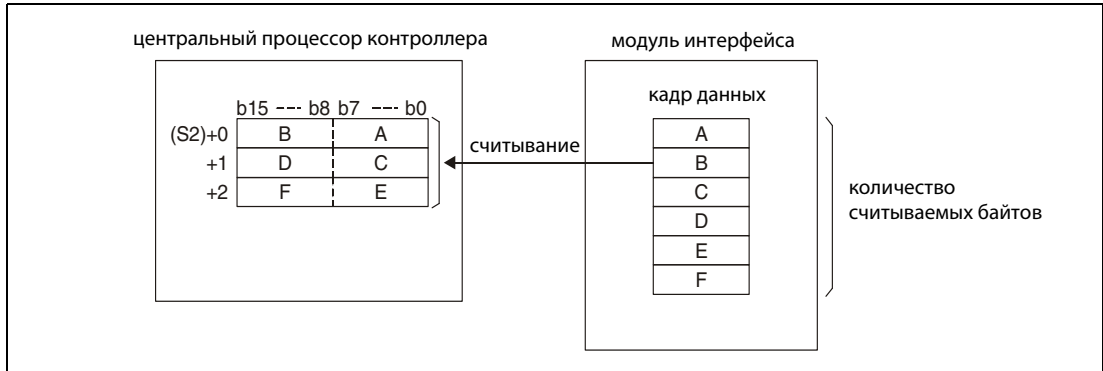
Операнд	Значение	Диапазон	Устанавливает	Тип данных	
Un	Головной адрес модуля интерфейса на монтажной шине (указываются только первые два разряда 3-разрядного адреса – например, головной адрес X/Y100 указывается в виде "10").	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит	
s1	Первый операнд области, содержащей информацию для выполнения команды				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(s1) + 0	фиктивный	Используется системой	0	—
	(s1) + 1	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды: 0000 <sub>H</sub> : безошибочная обработка Любое иное значение кроме 0000 <sub>H</sub> : При обработке возникла ошибка. Записанное значение представляет собой код ошибки.		система
	(s1) + 2	Используемый кадр данных	Номер кадра данных, определенного пользователем	от 1000 до 1199	пользователь
(s1) + 3	Количество считываемых байтов	Макс. количество байтов кадра данных, которое может быть сохранено в s2.	от 1 до 80		
	Количество считанных байтов	Количество действительно считанных байтов кадра данных	от 1 до 80	система	
s2	Первый операнд области, в котором сохраняются считанные данные.		пользователь система	адрес	
d	Битовый операнд, устанавливаемый на один цикл после выполнения команды GETE. По адресу (d) + 1 сигнализируется ошибочное завершение.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(d) + 0	Команда выполнена	Показывает завершение команды GETE. ВКЛ. : команда выполнена ВыКЛ.: команда не выполнена	—	система
(d) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды GETE. ВКЛ.: команда выполнена с ошибкой ВыКЛ.:команда выполнена без ошибок	—		

**Принцип действия**

**Считывание кадров данных, определенных пользователем**

**GETE Считывание данных**

Команда GETE считывает определенные пользователем кадры данных из модуля последовательного интерфейса и сохраняет их в центральном процессоре контроллера. Головной адрес модуля интерфейса указывается с помощью Up.

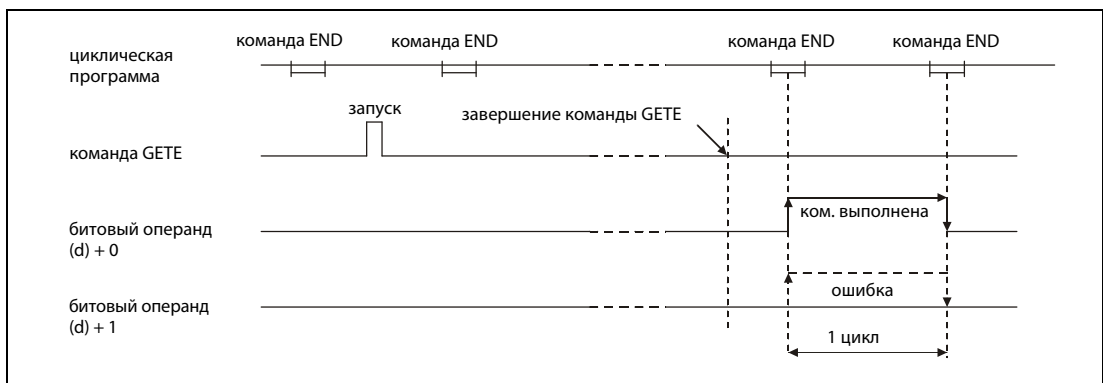


Во время выполнения команды GETE не может выполняться другая команда GETE. Выполнение команды PUTE также не возможно. Если во время обработки команды GETE запущена другая команда GETE или PUTE, система задерживает выполнение этих команд до полного завершения уже обрабатываемой команды GETE.

Завершено ли выполнение команды GETE, можно определить по состоянию битовых операндов (d) + 0 и (d) + 1:

- Битовый операнд (d) + 0 устанавливается при выполнении команды END того цикла, в котором была завершена команда GETE. При обработке следующей команды END указанный в (d) + 0 битовый операнд снова сбрасывается.
- Битовый операнд (d) + 1 показывает ошибку при выполнении команды GETE. При безошибочном выполнении этот битовый операнд остается сброшенным. При наличии ошибки операнд (d) + 1 устанавливается при выполнении команды END того цикла, в котором была завершена команда GETE. При обработке следующей команды END этот битовый операнд снова сбрасывается.

Ниже показана временная диаграмма выполнения команды GETE:



**Источники ошибок**

Если при выполнении команды GETE возникла ошибка, то устанавливается операнд (d) + 1 и в операнд (s1) + 1 записывается код ошибки. Коды ошибок подробно разъяснены в следующих руководствах:

- При коде ошибки до 4FFF<sub>H</sub> указания по устранению ошибки имеются в этом руководстве (глава 13).
- При коде ошибки, начиная с 7000<sub>H</sub>, более подробную информацию можно найти в руководстве по модулю интерфейса.



**Пример**

**GETE**

Следующая программа считывает из модуля QJ71C24 данные определенного пользователем кадра данных, которому присвоен номер 3E8<sub>H</sub>, и сохраняет эти данные в центральном процессоре "System Q", начиная с регистра данных D4. Модуль интерфейса занимает диапазон адресов от X/Y80 до X/Y9F.

● Редакторы по стандарту IEC

Релейно-контактная схема (GX IEC Developer)

преобразование запроса на считывание данных в импульс

стирание фиктивного операнда (s1) + 0

ввод номера кадра данных в (s1) + 1

считывание максимум 80 байтов

Стирается область данных, в которой сохраняются считанные данные.

считывание определенного пользователем кадра данных

Запрограммируйте в этом месте команды, которые должны выполняться, если команда GETE завершена без ошибок.

Запрограммируйте в этом месте команды, которые должны выполняться, если при выполнении команды GETE возникла ошибка.

---

Язык IL по IEC

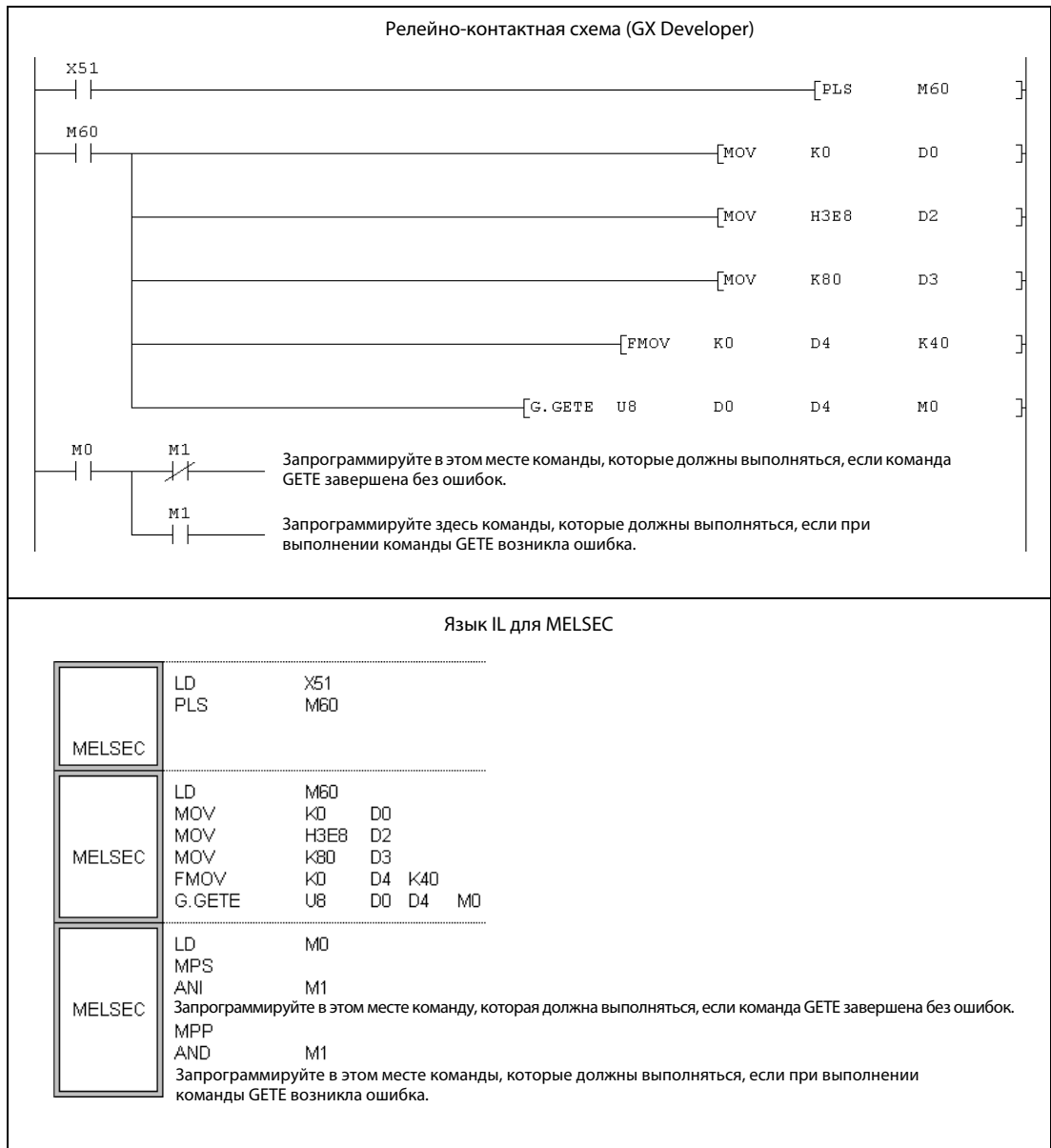
LD	PLS_M	X51			
			M60		
LD		M60			
MOV_M		0,	D0		
MOV_M		16#3E8,	D2		
MOV_M		80,	D3		
FMOV_M		0,	40,	D4	
GETE_M		8,	var_D0,	D4,	M0
LD		M0			
ANDN		M1			
В этом месте запрограммируйте команду, которая должна выполняться, если команда GETE завершена без ошибок.					
LD		M0			
AND		M1			
В этом месте запрограммируйте команду, которая должна выполняться, если при обработке команды GETE возникла ошибка.					

Операнды и команды разъяснены выше при рассмотрении примера программы на языке релейно-контактных схем.

**ПРИМЕЧАНИЕ**

В редакторах по стандарту IEC среды GX IEC Developer переменные должны быть объявлены в заголовке программного компонента (POU). Если переменные не были объявлены, то при проверке или компиляции программы возникает сообщение об ошибке. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

- Список инструкций MELSEC и релейно-контактная схема в среде GX Developer  
Операнды и команды разъяснены на предыдущей странице при рассмотрении примера программы на языке LD в среде GX IEC Developer.



### 11.1.3 PUTE, PUTEP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

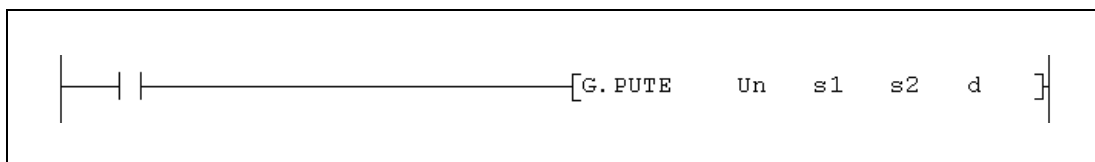
Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.ов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□\□□		Спец. модули U□\G□□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	—	SMO	
s2	—	●	●	—	—	—	—	—	—		
d	●	●	●	—	—	—	—	—	—		

GX IEC  
Developer



GX  
Developer



Переменные

Операнд	Значение		Диапазон	Устанавливает	Тип данных
Un	Головной адрес модуля интерфейса на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде "10".)		от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит
s1	Первый операнд области, содержащей информацию для выполнения команды.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(s1) + 0	Выбор: запись или стирание	Здесь выбирается, должен ли указанный в (s1) + 2 кадр данных быть стерт или установлен: • 1: запись данных • 3: стирание данных	1 или 3	пользователь
	(s1) + 1	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды. 0000 <sub>H</sub> : безошибочная обработка Любое иное значение кроме 0000 <sub>H</sub> : При обработке возникла ошибка. Записанное значение представляет собой код ошибки.	—	система
	(s1) + 2	Номер кадра данных	Номер кадра данных, определенного пользователем	от 1000 до 1199	пользователь
(s1) + 3	Количество записываемых байтов	Количество байтов, которое требуется передать в кадрах данных. Даже если данные просто стираются (т. е. (s1) + 0 = 3), укажите любое значение между 1 и 80.	от 1 до 80		
s2	Первый операнд области, в которой сохранены передаваемые данные.			пользователь	адрес
d	Битовый операнд, устанавливаемый на один цикл после выполнения команды PUTE. По адресу (d)+1 сигнализируется ошибочное завершение.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(d) + 0	Команда выполнена	Показывает завершение команды PUTE. ВКЛ.: команда выполнена ВЫКЛ.: команда не выполнена	—	система
(d) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды PUTE. ВКЛ.: команда выполнена с ошибкой ВЫКЛ.: команда выполнена без ошибок	—		

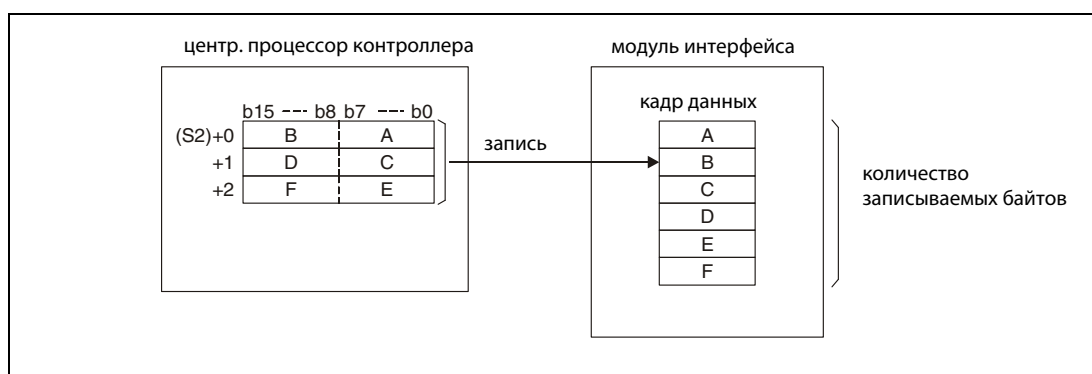
**Принцип действия****Установка или стирание определенных пользователем кадров данных****PUTE Команда установки и стирания**

Команда PUTE записывает определенные пользователем кадры данных в модуль последовательного интерфейса или стирает данные в этом модуле. Головной адрес модуля интерфейса указывается с помощью  $U_n$ .

**Запись определенного пользователем кадра данных**

Для записи определенного пользователем кадра данных указанный в  $(s1) + 0$  операнд должен иметь значение "1". Данные из области операндов, указанной в  $s2$ , передаются в модуль интерфейса. Каждый из этих операндов может принять два байта. Таким образом, требуемое количество операндов соответствует половине количества байтов данных.

Например, если в кадр данных вводятся шесть байтов, то за операндом  $s2$  требуется зарезервировать еще два операнда:

**Стирание определенного пользователем кадра данных**

Для стирания пользовательского кадра данных, указанного в  $(s1) + 2$ , операнд, указанный в  $(s1) + 0$ , должен иметь значение "3".

При стирании длина данных  $[(s1) + 3]$  и область операндов  $s2$  не учитываются, однако для правильного выполнения команды они все-таки должны быть указаны. Для операнда, указанного в  $(s1) + 3$ , введите любое значение между 1 и 80, а для  $s2$  выберите фиктивный операнд.

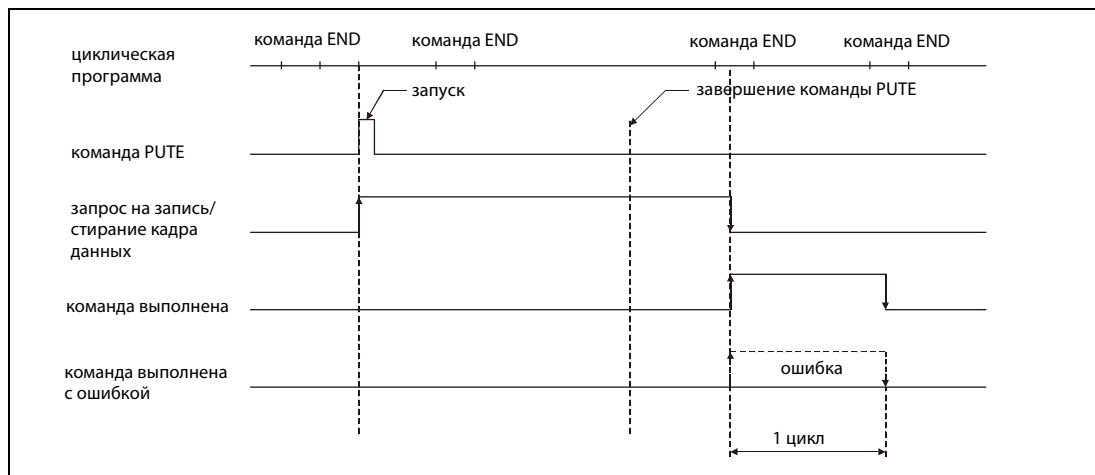
**Условия выполнения**

Во время выполнения команды PUTE не может выполняться другая команда PUTE. Выполнение команды GETE также не возможно. Если во время обработки команды PUTE запущена другая команда PUTE или команда GETE, система задерживает выполнение этих команд до полного завершения уже выполняемой команды PUTE.

Завершено ли выполнение команды PUTE, можно определить по состоянию битовых операндов  $(d) + 0$  и  $(d) + 1$ :

- Битовый операнд  $(d) + 0$  устанавливается при выполнении команды END того цикла, в котором была завершена команда PUTE. При обработке следующей команды END указанный в  $d + 0$  битовый операнд снова сбрасывается.
- Битовый операнд  $(d)+1$  показывает ошибку при выполнении команды PUTE. При безошибочном выполнении этот битовый операнд остается сброшенным. При наличии ошибки операнд  $(d)+1$  устанавливается при выполнении команды END того цикла, в котором была завершена команда PUTE. При обработке следующей команды END указанный в  $(d) + 1$  битовый операнд снова сбрасывается.

На рисунке ниже изображена временная диаграмма выполнения команды PUTE:



**Источники ошибок**

Если при выполнении команды PUTE возникла ошибка, то устанавливается операнд (d) + 1 и в операнд (s1) + 1 вводится код ошибки. Коды ошибок подробно разъяснены в следующих руководствах:

- При коде ошибки до 4FFF<sub>H</sub> указания по устранению ошибки имеются в этом руководстве (глава 13).
- При коде ошибки, начиная с 7000<sub>H</sub>, более подробную информацию можно найти в руководстве по модулю интерфейса.

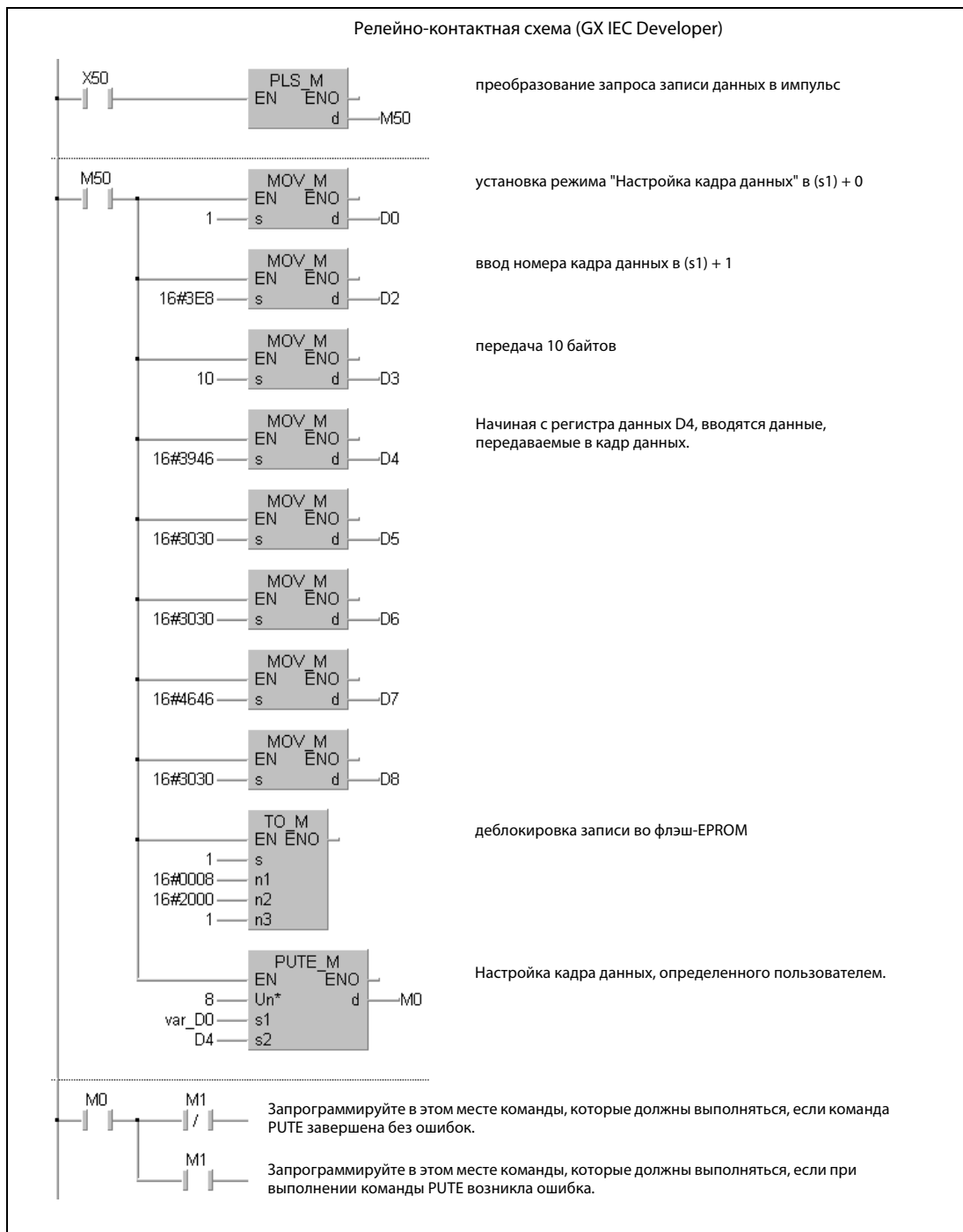
**Пример** PUTE

Следующая программа передает настройки пользовательского кадра данных № 3E8H. Используемый модуль интерфейса QJ71C24 занимает диапазон адресов от X/Y80 до X/Y9F.

**ПРИМЕЧАНИЕ**

В редакторах по стандарту IEC среды GX IEC Developer переменные должны быть объявлены в заголовке программного компонента (POU). Если переменные не были объявлены, то при проверке или компилировании программы возникает сообщение об ошибке. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

- Редакторы по стандарту IEC

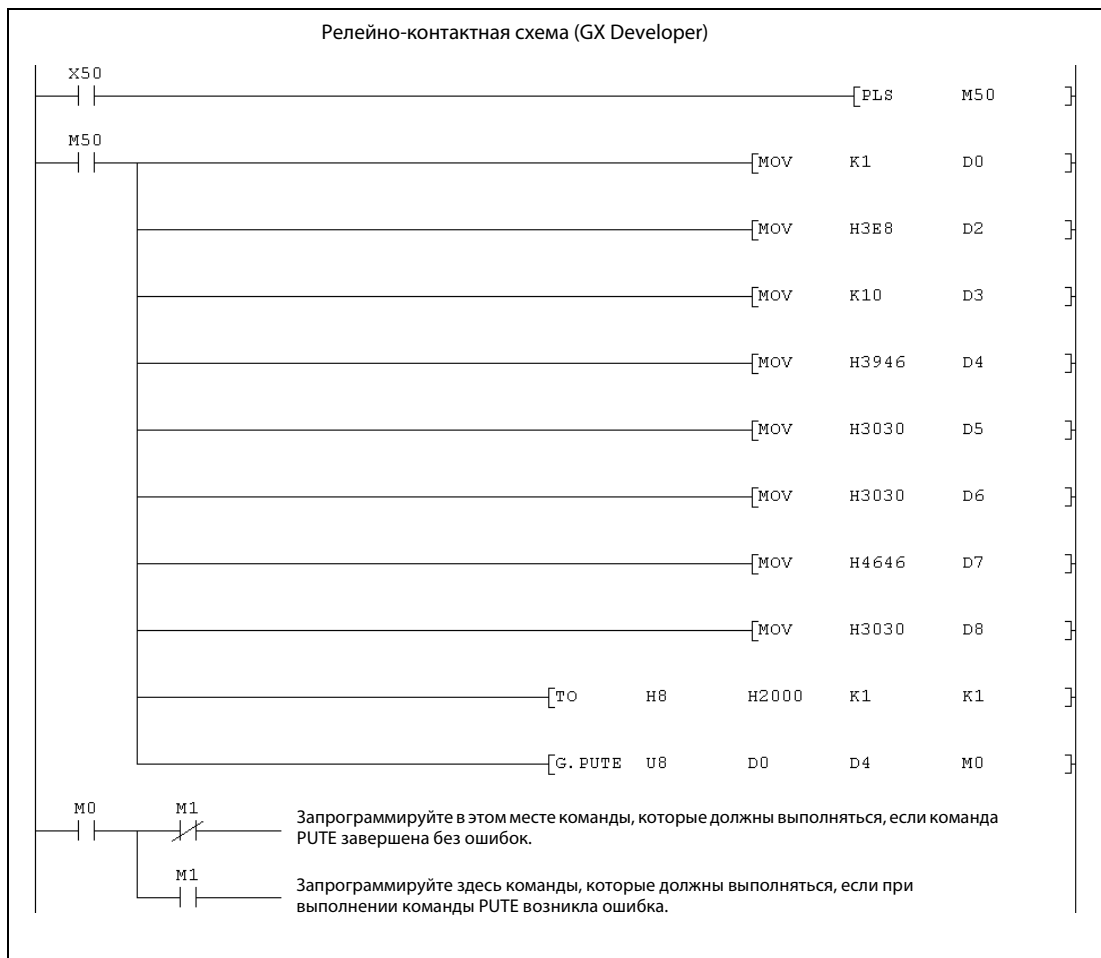


Язык IL по стандарту IEC

	LD	X50			
	PLS_M	M50			
	LD	M50			
	MOV_M	1,	D0		
	MOV_M	16#3E8,	D2		
	MOV_M	10,	D3		
	MOV_M	16#3946,	D4		
	MOV_M	16#3030,	D5		
	MOV_M	16#3030,	D6		
	MOV_M	16#4646,	D7		
	MOV_M	16#3030,	D8		
	TO_M	1,	16#0008,	16#2000,	1
	PUTE_M	8,	var_DD,	D4,	M0
	LD	M0			
	ANDN	M1			
	Запрограммируйте в этом месте команду, которая должна выполняться, если команда PUTE завершена без ошибок.				
	LD	M0			
	AND	M1			
	Запрограммируйте в этом месте команду, которая должна выполняться, если при обработке команды PUTE возникла ошибка.				

Операнды и команды разъяснены при рассмотрении примера программы на языке LD.

- Список инструкций MELSEC и релейно-контактная схема в среде GX Developer  
 Операнды и команды разъяснены на предыдущей странице при рассмотрении примера программы на языке LD в среде GX IEC Developer.





Язык IL для MELSEC

MELSEC	LD	X50			
	PLS	M50			
MELSEC	LD	M50			
	MOV	K1	D0		
	MOV	H3E8	D2		
	MOV	K10	D3		
	MOV	H3946	D4		
	MOV	H3030	D5		
	MOV	H3030	D6		
	MOV	H4646	D7		
	MOV	H3030	D8		
	TO	H8	H2000	K1	K1
G.PUTE	U8	D0	D4	M0	
MELSEC	LD	M0			
	MPS				
	ANI	M1			
	Запрограммируйте в этом месте команду, которая должна выполняться, если команда PUTE завершена без ошибок.				
	MPP				
MELSEC	AND	M1			
	Запрограммируйте в этом месте команды, которые должны выполняться, если при выполнении команды PUTE возникла ошибка.				

11.1.4 PRR, PRRP

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

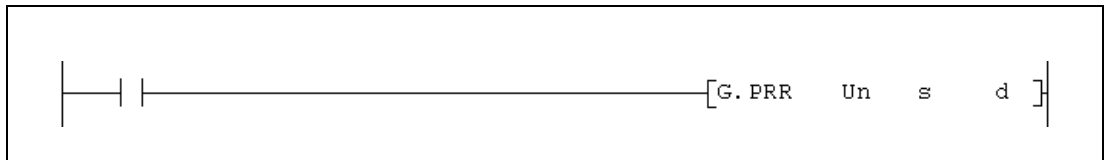
Операнды  
MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользov.)		Регистры файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	—	SM0	
d1	●	●	●	—	—	—	—	—	—		

GX IEC  
Developer



GX  
Developer



## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных	
Un	Головной адрес модуля интерфейса на монтажной шине. (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде "10".)	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит	
s	Первый операнд области, содержащей информацию для выполнения команды.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(s) + 0	Канал передачи	Указание интерфейса, через который передаются данные 1: интерфейс 1 (CH1) 2: интерфейс 2 (CH2)	1 или 2	пользователь
	(s) + 1	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды. 0000 <sub>H</sub> : Безошибочная обработка Любое иное значение кроме 0000 <sub>H</sub> : При обработке возникла ошибка. Записанное значение представляет собой код ошибки.	—	система
	(s) + 2	Присоединение CR/LF	Выбор, должны ли к переданным данным добавляться коды CR/LF 0: не добавлять CR/LF 1: добавлять CR/LF	0 или 1	пользователь
	(s) + 3	Указатель данных	Указатель на первый адрес области операндов, в которой сохранены передаваемые данные.	от 1 до 100	
(s) + 4	Количество передаваемых кадров данных	Указание, сколько кадров данных требуется передать.	от 1 до 100		
d	Битовый операнд, устанавливаемый на один цикл после выполнения команды PRR. По адресу (d) + 1 сигнализируется ошибочное завершение.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(d) + 0	Команда выполнена	Показывает завершение команды PRR. ВКЛ. : команда выполнена Выкл.: команда не выполнена	—	система
(d) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды PRR. ВКЛ.: команда выполнена с ошибкой Выкл.: команда выполнена без ошибок	—		

**Принцип действия**

**Передача определенных пользователем кадров данных**

**PRR Команда передачи**

Команда PRR передает данные на основе определенных пользователем кадров данных в модуль интерфейса, указанный в Un. Настройки для выполнения команды заложены начиная с операнда, указанного в s. Перед выполнением команды PRR содержимое кадров данных должно быть записано в модуль интерфейса.

Во время выполнения команды PRR, для того же интерфейса не могут выполняться следующие команды:

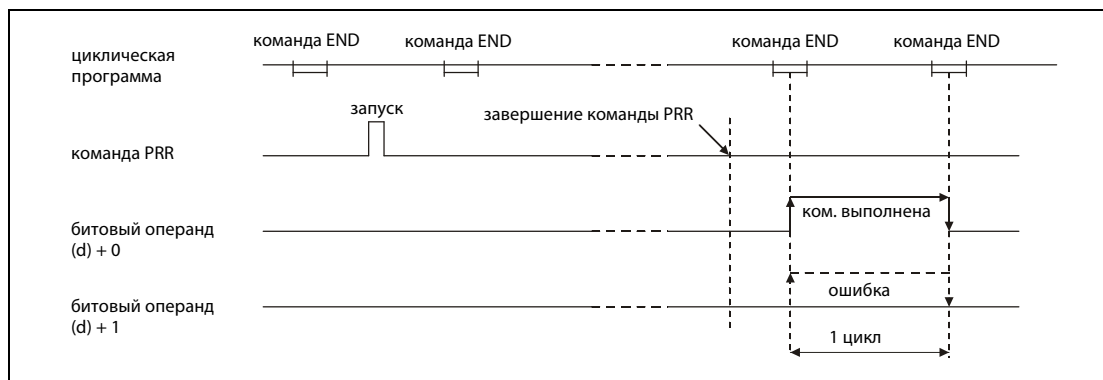
OUTPUT, ONDEMAND, BIDOUT, а также другая команда PRR

Если во время обработки команды PRR запущена одна из этих команд, система задерживает их выполнение до полного завершения выполняемой команды PRR.

Завершено ли выполнение команды PRR, можно определить по состоянию битовых операндов (d) + 0 и (d) + 1:

- Битовый операнд (d) + 0 устанавливается при выполнении команды END того цикла, в котором была завершена команда PRR. При обработке следующей команды END указанный в (d) + 0 битовый операнд снова сбрасывается.
- Битовый операнд (d) + 1 показывает ошибку при выполнении команды PRR. При безошибочном выполнении этот битовый операнд остается сброшенным. При наличии ошибки операнд (d) + 1 устанавливается при выполнении команды END того цикла, в котором была завершена команда PRR. При обработке следующей команды END этот битовый операнд снова сбрасывается.

На рисунке ниже показана временная диаграмма выполнения команды PRR:



**Источники ошибок**

Если при выполнении команды PRR возникла ошибка, то устанавливается операнд (d) + 1 и в операнд s + 1 вводится код ошибки. Коды ошибок подробно разъяснены в следующих руководствах:

- При коде ошибки до 4FFF<sub>H</sub> указания по устранению ошибки имеются в этом руководстве (глава 13).
- При коде ошибки, начиная с 7000<sub>H</sub>, более подробную информацию можно найти в руководстве по модулю интерфейса.

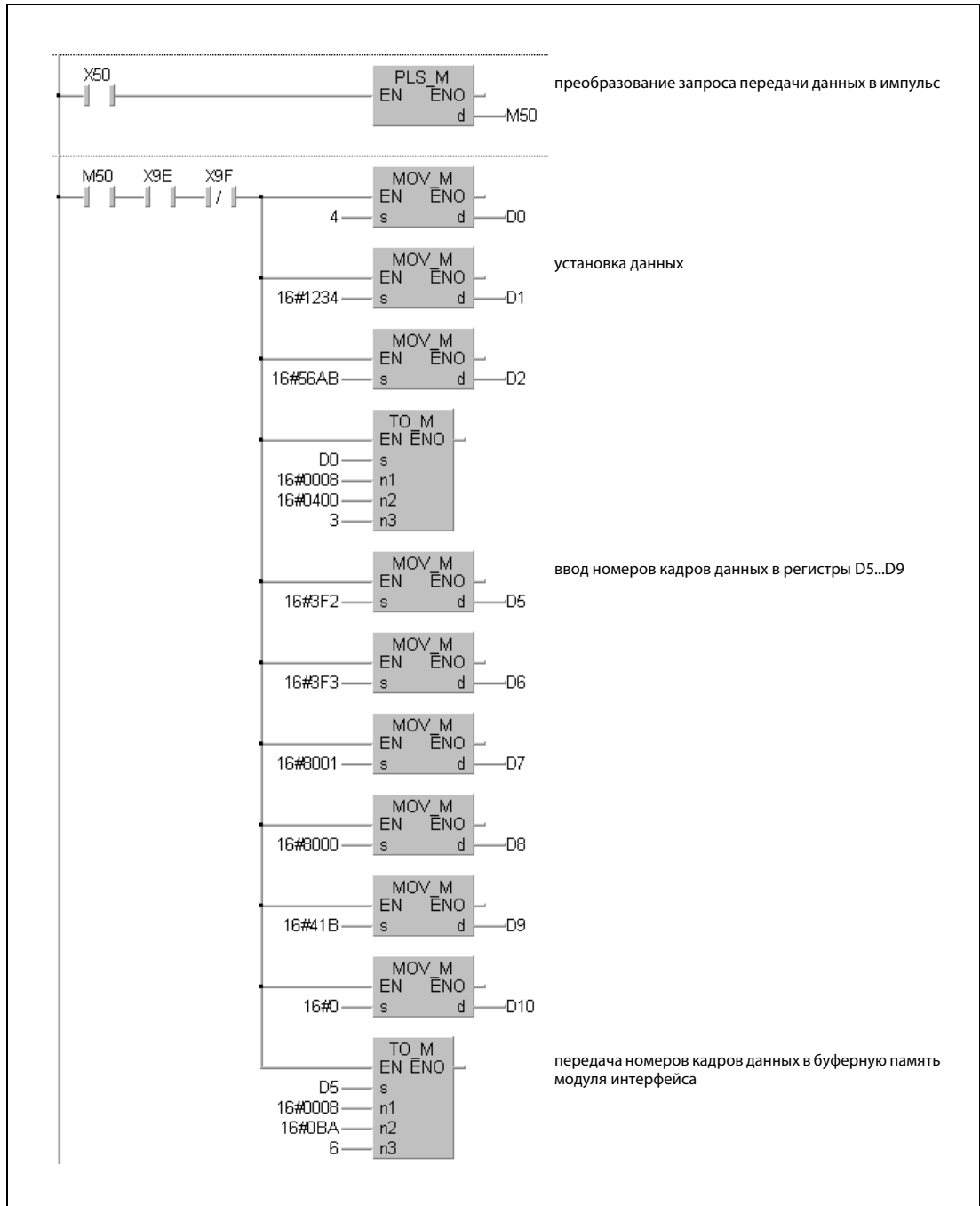
**Пример** PRR

Эта программа передает данные и первые пять пользовательских кадров данных. Используемый модуль интерфейса QJ71C24 занимает диапазон адресов от X/Y80 до X/Y9F. Используются регистры данных, указанные в следующей таблице:

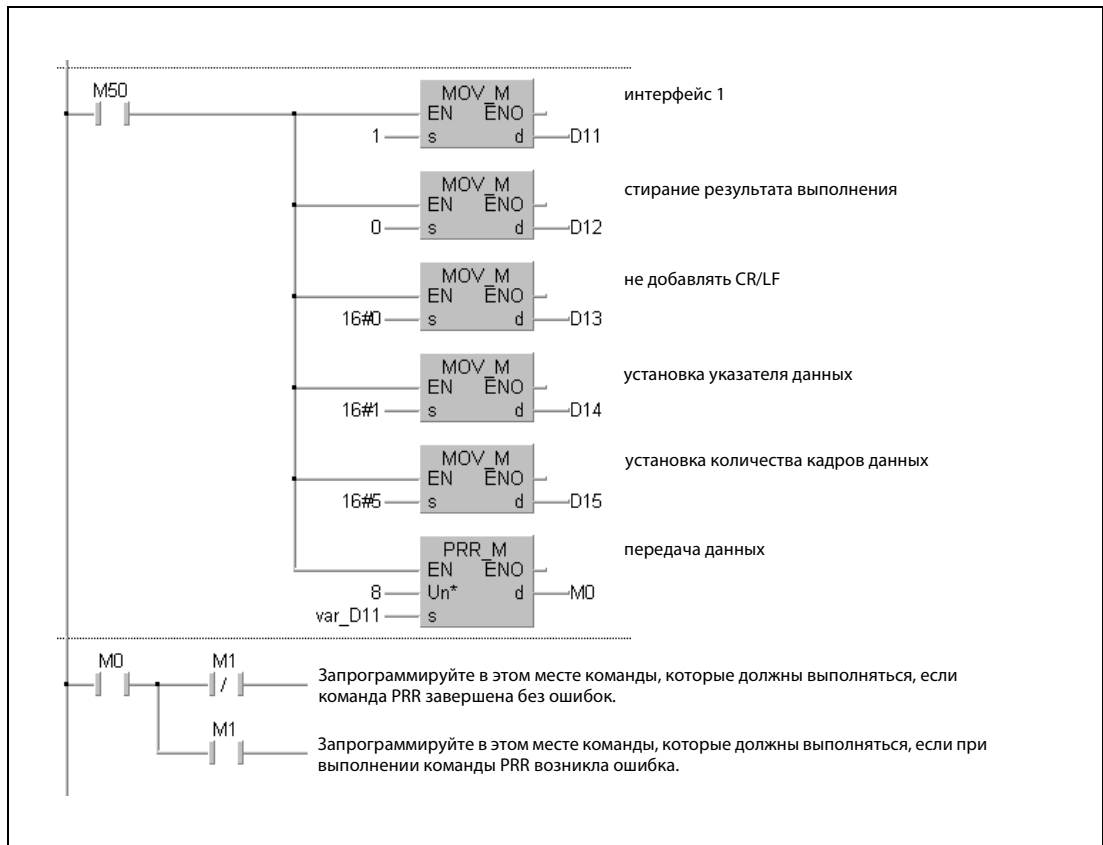
Регистр данных	Содержимое	Значение	
D0	0004 <sub>H</sub>	Количество передаваемых байтов	
D1	3412 <sub>H</sub>	Передаваемые данные	
D2	AB56 <sub>H</sub>		
D5	03F2 <sub>H</sub>	Номера кадров данных	
D6	03F3 <sub>H</sub>		
D7	8001 <sub>H</sub>		
D8	8000 <sub>H</sub>		
D9	041B <sub>H</sub>		
D10	0000 <sub>H</sub>		
D11	0001 <sub>H</sub>	(s) + 0	Интерфейс: CH1
D12	0000 <sub>H</sub> или код ошибки	(s) + 1	Результат выполнения
D13	0000 <sub>H</sub>	(s) + 2	Не добавлять CR/LF.
D14	0001 <sub>H</sub>	(s) + 3	Указатель данных
D15	0005 <sub>H</sub>	(s) + 4	Количество передаваемых кадров данных

**ПРИМЕЧАНИЕ** В редакторах по стандарту IEC среды GX IEC Developer переменные должны быть объявлены в заголовке программного компонента (POU). Если переменные не были объявлены, то при проверке или компилировании программы возникает сообщение об ошибке. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

- Редакторы по стандарту IEC  
Релейно-контактная схема в среде GX IEC Developer (часть 1)



Релейно-контактная схема в среде GX IEC Developer (продолжение)

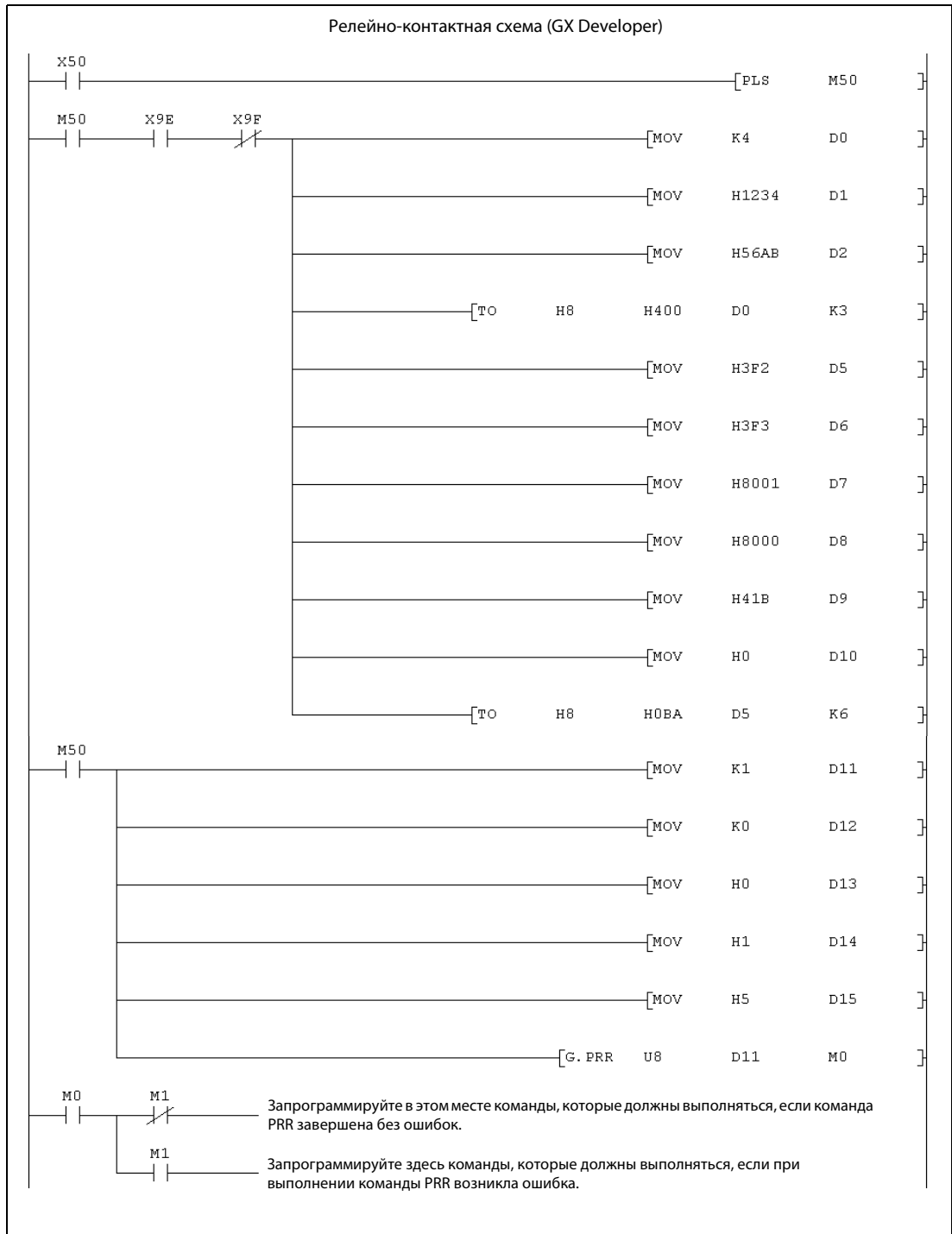


Язык IL по IEC

LD	X50		
PLS_M	M50		
-----			
LD	M50		
AND	X9E		
ANDN	X9F		
MOV_M	4,	D0	
MOV_M	16#1234,	D1	
MOV_M	16#56AB,	D2	
TO_M	D0,	16#0008,	16#0400, 3
MOV_M	16#3F2,	D5	
MOV_M	16#3F3,	D6	
MOV_M	16#8001,	D7	
MOV_M	16#8000,	D8	
MOV_M	16#41B,	D9	
MOV_M	16#0,	D10	
TO_M	D5,	16#0008,	16#0BA, 6
-----			
LD	M50		
MOV_M	1,	D11	
MOV_M	0,	D12	
MOV_M	16#0,	D13	
MOV_M	16#1,	D14	
MOV_M	16#5,	D15	
PRR_M	8,	var_D11,	M0
-----			
LD	M0		
ANDN	M1		
Запрограммируйте в этом месте команду, которая должна выполняться, если команда PRR завершена без ошибок.			
LD	M0		
AND	M1		
Запрограммируйте в этом месте команду, которая должна выполняться, если при обработке команды PRR возникла ошибка.			

Операнды и команды разъяснены при рассмотрении примера программы на языке LD.

- Список инструкций MELSEC и релейно-контактная схема в среде GX Developer  
 Операнды и команды разъяснены при рассмотрении примера программы на языке LD в среде GX IEC Developer.





Язык IL для MELSEC

MELSEC	LD X50 PLS M50				
MELSEC	LD M50 AND X9E ANI X9F MOV K4 D0 MOV H1234 D1 MOV H56AB D2 TO H8 H400 D0 K3 MOV H3F2 D5 MOV H3F3 D6 MOV H8001 D7 MOV H8000 D8 MOV H41B D9 MOV H0 D10 TO H8 H0BA D5 K6				
MELSEC	LD M50 MOV K1 D11 MOV K0 D12 MOV H0 D13 MOV H1 D14 MOV H5 D15 G.PRR U8 D11 M0				
MELSEC	LD M0 MPS ANI M1 MPP AND M1	Запрограммируйте в этом месте команду, которая должна выполняться, если команда PRR завершена без ошибок. Запрограммируйте в этом месте команды, которые должны выполняться, если при выполнении команды PRR возникла ошибка.			

## 11.2 Команды для модулей Profibus/DP

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту IEC
Считывание данных из буферной памяти модулей Profibus/DP	G.BBLKRD	BBLKRD_M
	GP.BBLKRD	BBLKRD_DP_M
Запись данных в буферную память модулей Profibus/DP	G.BBLKWR	BBLKWR_M
	GP.BBLKWR	BBLKWR_DP_M

**11.2.1 BBLKRD, BBLKRDП**

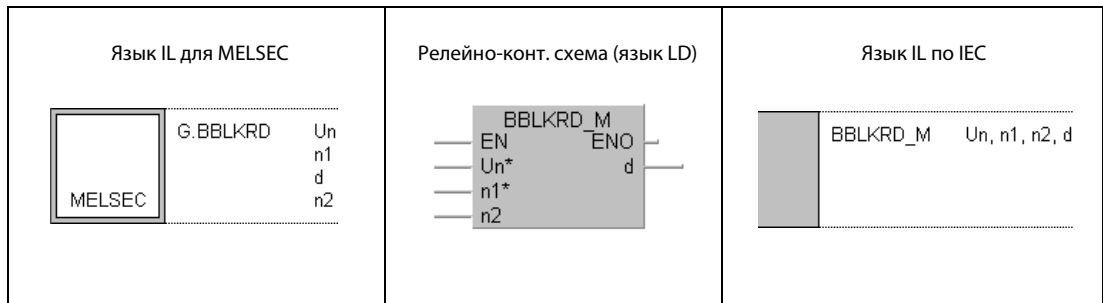
**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

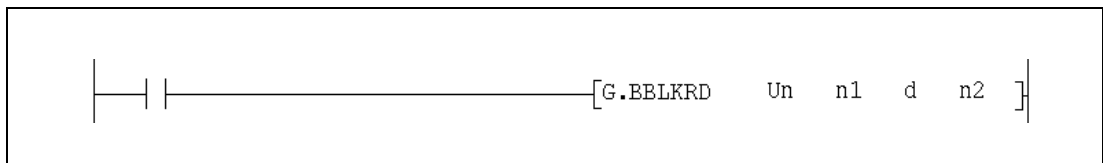
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.ов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
n1	—	●	●	—	—	—	—	●	—	SMO	
d	—	●	●	—	—	—	—	—	—		
n2	—	●	●	—	—	—	—	●	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

Операнд	Значение	Тип данных
Un	Головной адрес модуля Profibus на монтажной шине	BIN, 16 бит
n1	Начальный адрес в буферной памяти модуля Profibus, с которого считываются данные.	
d	Начальный адрес области операндов, в котором сохраняются считанные данные.	обозначение операнда
n2	Количество считываемых данных	BIN, 16 бит

**Принцип действия**

**Считывание данных из буферной памяти модуля Profibus**

**BBLKRD / BBLKRDП**

**Считывание данных**

Команда BBLKRD служит для считывания содержимого буферной памяти модулей Profibus QJ71PB92D и QJ71PB93D. Считанные данные (например, данные выходов модуля) передаются в центральный процессор в виде взаимосвязанной общности данных.

Модуль QJ71PB93 необходимо подготавливать к команде BBLKRD путем установки выходного сигнала Y0A. Если вслед за этим модуль Profibus устанавливает вход X0A, можно выполнять команду BBLKRD. После считывания буферной памяти выходной сигнал Y0A следует снова сбрасывать.

Диапазоны значений и указание операндов:

- Un (головной адрес модуля Profibus на монтажной шине): от 0 до FF<sub>H</sub>  
(Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде "10"<sub>H</sub>.)
- n1 (начальный адрес в буферной памяти): указанный адрес должен существовать.
- d (начальный адрес целевой области операндов): указанный здесь операнд должен существовать.
- n2 (количество считываемых данных)  
В случае QJ71PB92D: от 1 до 960 слов (от 1 до 3C0<sub>H</sub>)  
В случае QJ71PB93D: от 1 до 122 слов (от 1 до 7A<sub>H</sub>)

**ПРИМЕЧАНИЯ**

*В одном и том же программном цикле разрешается выполнять только одну команду BBLKRD.*

*Команды BBLKRD и BBLKWR (глава 11.2.2) работают независимо друг от друга.*

*При использовании команды BBLKRD увеличивается время задержки передачи.*

*Если в параметрах мастер-станции не указан выходной модуль, команда BBLKRD не выполняется.*

**Источники ошибок**

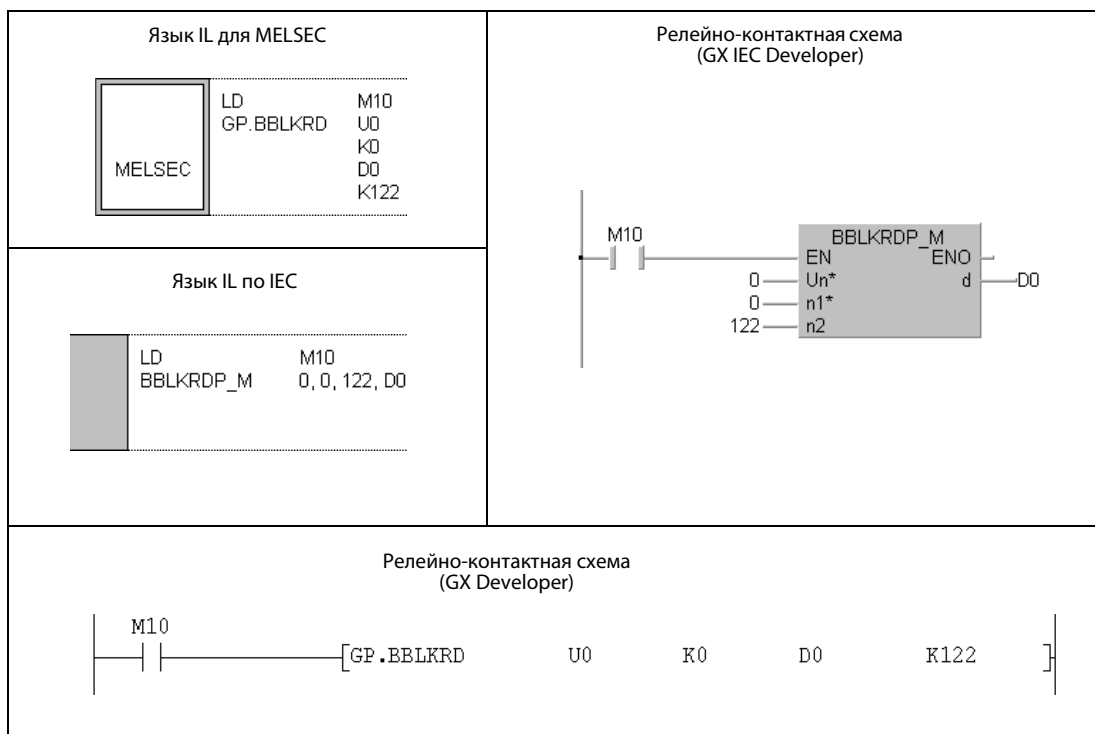
В следующих случаях возникает ошибка обработки, устанавливается флаг ошибки SM0 и в регистр SD0 записывается код ошибки:

- Настроенное значение находится вне допустимого диапазона (код ошибки: 4101).
- При сложении указанного в n1 начального адреса с указанным в n2 количеством слов данных превышает диапазон адресов буферной памяти (код ошибки 4101).
- Указанное в n2 количество слов данных превышает имеющуюся область операндов для считанных данных, начальный адрес которой указан в d (код ошибки 4101).

**Пример**

**BBLKRDП**

При установке маркера M10 из модуля Profibus QJ71PB93D с головным адресом X/Y0 считываются 122 слова, начиная с адреса 0 буферной памяти. Эти данные сохраняются в центральном процессоре, начиная с регистра данных D0.



### 11.2.2 BBLKWR, BBLKWRP

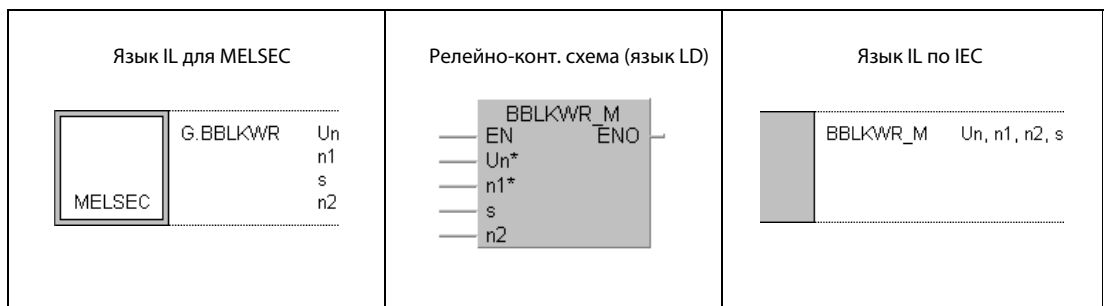
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

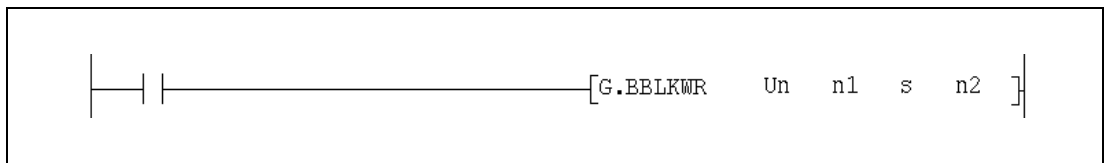
Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
n1	—	●	●	—	—	—	—	●	—	SM0	
s	—	●	●	—	—	—	—	—	—		
n2	—	●	●	—	—	—	—	●	—		

GX IEC Developer



GX Developer



Переменные

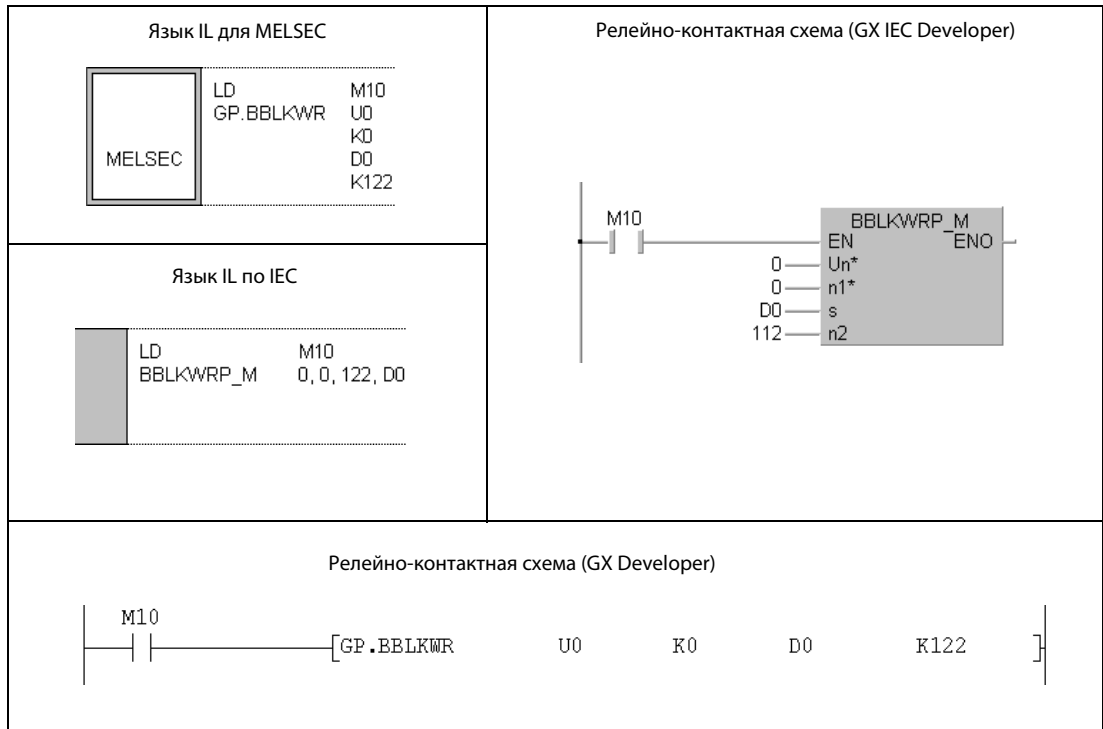
Операнд	Значение	Тип данных
Un	Головной адрес модуля Profibus на монтажной шине	BIN, 16 бит
n1	Начальный адрес в буферной памяти модуля Profibus, в которую записываются данные.	
s	Начальный адрес области операндов, в которой сохранены данные, передаваемые в модуль Profibus.	обозначение операнда
n2	Количество передаваемых данных	BIN, 16 бит

<b>Принцип действия</b>	<b>Запись данных в буферную память модуля Profibus BBLKWR/BBLKWRP                      Запись данных</b>
	<p>Команда BBLKWR записывает данные в буферную память модулей Profibus/DP QJ71PB92D и QJ71PB93D. Во время записи предотвращается разбивка данных.</p> <p>Модуль QJ71PB93 должен быть подготовлен к команде BBLKWR путем установки выходного сигнала Y0B. Если вслед за этим модуль Profibus устанавливает вход X0B (в качестве подтверждения), можно выполнять команду BBLKWR. После передачи данных выходной сигнал Y0B следует снова сбрасывать.</p> <p>Диапазоны значений и указание операндов:</p> <ul style="list-style-type: none"><li>● Un (головной адрес модуля Profibus на монтажной шине): от 0 до FF<sub>H</sub> (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде "10"<sub>H</sub>.)</li><li>● n1 (начальный адрес в буферной памяти): указанный адрес должен существовать.  В модуле QJ71PB93 начальный адрес имеет смещение 100<sub>H</sub>. Поэтому при указании n1 от требуемого начального адреса необходимо вычесть значение 100<sub>H</sub>. Например, начальный адрес 100<sub>H</sub> указывается в виде "0"<sub>H</sub>, а начальный адрес 120<sub>H</sub> – в виде "20"<sub>H</sub>.</li><li>● s (начальный адрес исходной области операндов): указанный здесь операнд должен существовать.</li><li>● n2 (количество считываемых данных) В случае модуля QJ71PB92D: от 1 до 960 слов (от 1 до 3C0<sub>H</sub>) В случае модуля QJ71PB93D: от 1 до 122 слов (от 1 до 7A<sub>H</sub>)</li></ul>
<b>ПРИМЕЧАНИЯ</b>	<p><i>В одном и том же программном цикле разрешается выполнять только одну команду BBLKWR. Команды BBLKWR и BBLKRD (глава 11.2.1) работают независимо друг от друга.</i></p> <p><i>При использовании команды BBLKWR увеличивается время задержки для передачи.</i></p> <p><i>Если в параметрах мастер-станции не указан входной модуль, команда BBLKWR не выполняется.</i></p>
<b>Источники ошибок</b>	<p>В следующих случаях возникает ошибка обработки, устанавливается флаг ошибки SM0 и в регистр SDO записывается код ошибки:</p> <ul style="list-style-type: none"><li>● Настроенное значение находится вне допустимого диапазона (код ошибки: 4101).</li><li>● При сложении указанного в n1 начального адреса с указанным в n2 количеством слов данных превышает диапазон адресов буферной памяти (код ошибки 4101).</li><li>● Указанное в n2 количество слов данных превышает имеющуюся область операндов для передаваемых данных, начальный адрес которой указан в s (код ошибки 4101).</li></ul>

**Пример**

**BBLKWRP**

После установки маркера M10 122 слова из области операндов D0...D121 центрального процессора передаются во входную область подчиненного модуля Profibus/DP QJ71PB93D. Входная область начинается с адреса в 100<sub>H</sub> буферной памяти. Учитывайте, что в этом случае начальный адрес в n1 указывается как "0<sub>H</sub>". Модуль Profibus/DP имеет головной адрес X/Y0.





## 11.3 Команды для модулей Ethernet

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту IEC
Считывание данных из фиксированных буферов	ZP.BUFRCV	BUFRCV_M
	Z.BUFRCVS	BUFRCVS_M
Передача данных в фиксированные буферы	ZP.BUFSND	BUFSND_M
Установление (открытие) связи	ZP.OPEN	OPEN_M
Закрытие связи	ZP.CLOSE	CLOSE_M
Стирание сообщений об ошибках из буфера	ZP.ERRCLR	ERRCLR_M
Считывание кода ошибки	ZP.ERRRD	ERRRD_M
Инициализация модуля Ethernet	ZP.UINI	UINI_M

### 11.3.1 BUFRCV

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

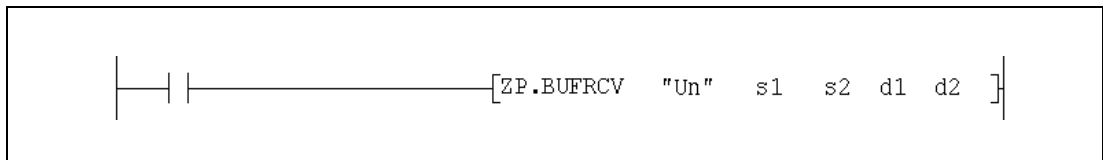
Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	●	—		
s2	—	●	●	—	—	—	—	—	—		
d1	—	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		

GX IEC  
Developer



GX  
Developer



## Переменные

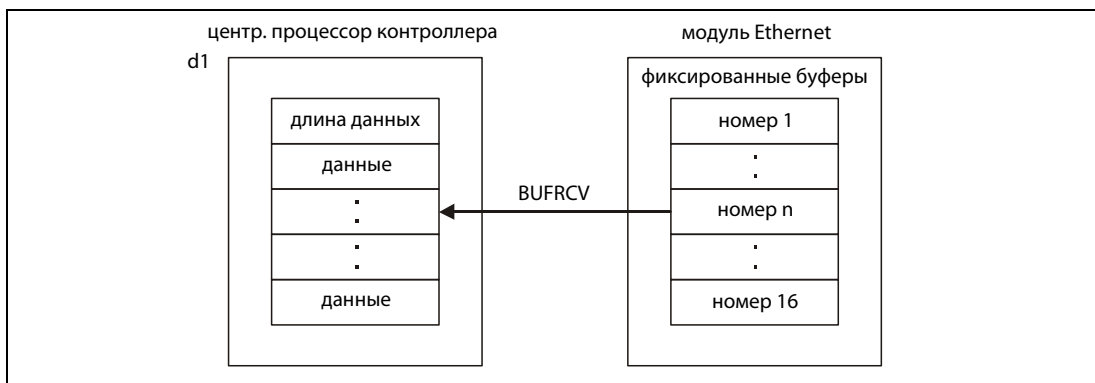
Операнд	Значение	Диапазон	Устанавливает	Тип данных	
"Un"	Головной адрес модуля Ethernet на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде "U10".)	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит	
s1	Номер соединения	от 1 до 16			
s2	Первый операнд области, содержащей информацию для выполнения команды				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(s2) + 0	системная область	Используется системой	—	система
(s2) + 1	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды: 0000 <sub>H</sub> : безошибочная обработка Любое иное значение кроме 0000 <sub>H</sub> : При обработке возникла ошибка. Записанное значение представляет собой код ошибки, разъяснение которого можно найти либо в руководстве по модулю Ethernet, либо в глава 13 этого руководства.			
d1	Первый операнд области, в которой сохраняются принятые данные.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(d1) + 0	Длина принятых данных	С процедурой (двоичные данные): Количество данных, считанных из фиксированного буфера	от 1 до 1017 слов	система
			С процедурой (данных в формате ASCII): Количество данных, считанных из фиксированного буфера.	от 1 до 508 слов	
Без процедуры (двоичные данные): Количество данных, считанных из фиксированного буфера.			от 1 до 2016 байтов		
от (d1) + 1 до (d1) + n	Принятые данные	В этой области данные, считанные из фиксированного буфера, поочередно записываются в возрастающей последовательности.	—		
d2	Битовый операнд, устанавливаемый на один цикл после выполнения команды BUFRCV. С помощью (d2) + 1 сигнализируется ошибочное завершение.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(d2) + 0	Команда выполнена	Показывает завершение команды BUFRCV. Вкл. : команда выполнена Выкл.: команда не выполнена	—	система
(d2) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды BUFRCV. Вкл.: команда выполнена с ошибкой Выкл.: команда выполнена без ошибок	—		

**Принцип действия**

**Считывание данных из фиксированных буферов (команда вызывается в основной программе)**

**BUFRCV Считывание данных**

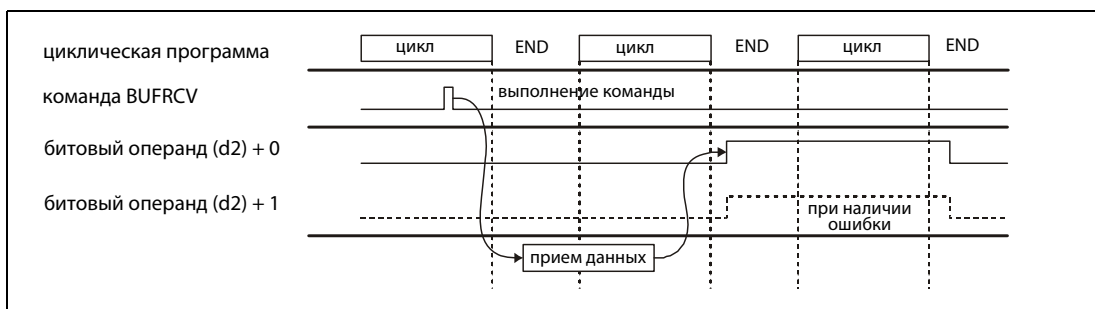
Команда BUFRCV считывает из модуля Ethernet данные, которые были посланы внешней станцией в модуль Ethernet при коммуникации с использованием фиксированного буфера, и сохраняет их в центральном процессоре контроллера. В отличие от команды BUFRCVS, команда BUFRCV применяется в основной программе. С помощью d1 указывается, где следует сохранить данные:



Завершено ли выполнение команды BUFRCV, можно определить по состоянию битовых операндов (d2) + 0 и (d2) + 1:

- Битовый операнд (d2) + 0 устанавливается при выполнении команды END того цикла, в котором команда BUFRCV была завершена. При обработке следующей команды END этот битовый операнд снова сбрасывается.
- Битовый операнд (d2) + 1 показывает ошибку при выполнении команды BUFRCV. При безошибочном выполнении этот битовый операнд остается сброшенным. При наличии ошибки операнд, указанный в (d2) + 1, устанавливается при выполнении команды END того цикла, в котором была завершена команда BUFRCV. При обработке следующей команды END битовый операнд, указанный в (d2) + 1, снова сбрасывается.

На рисунке ниже показана диаграмма изменения сигналов при выполнении команды BUFRCV:



Выполнение команды BUFRCV возможно только в том случае, если модуль Ethernet сигнализирует о готовности к приему данных. Для этого по адресу 5005<sub>H</sub> буферной памяти модуля Ethernet устанавливается один бит. Здесь зарезервировано по одному биту для каждого из 16 возможных соединений.

**ПРИМЕЧАНИЕ**

Принятые данные одного и того же соединения невозможно одновременно считывать с помощью команд BUFRCV (для основных программ) и BUFRCVS (для программ прерывания).

**Источники ошибок**

Если команда BUFRCV выполнена с ошибками, то устанавливается операнд (d2) + 1 и соответствующий код ошибки записывается в (s2) + 1. Подробные сведения об отдельных кодах ошибок имеются в следующих руководствах:

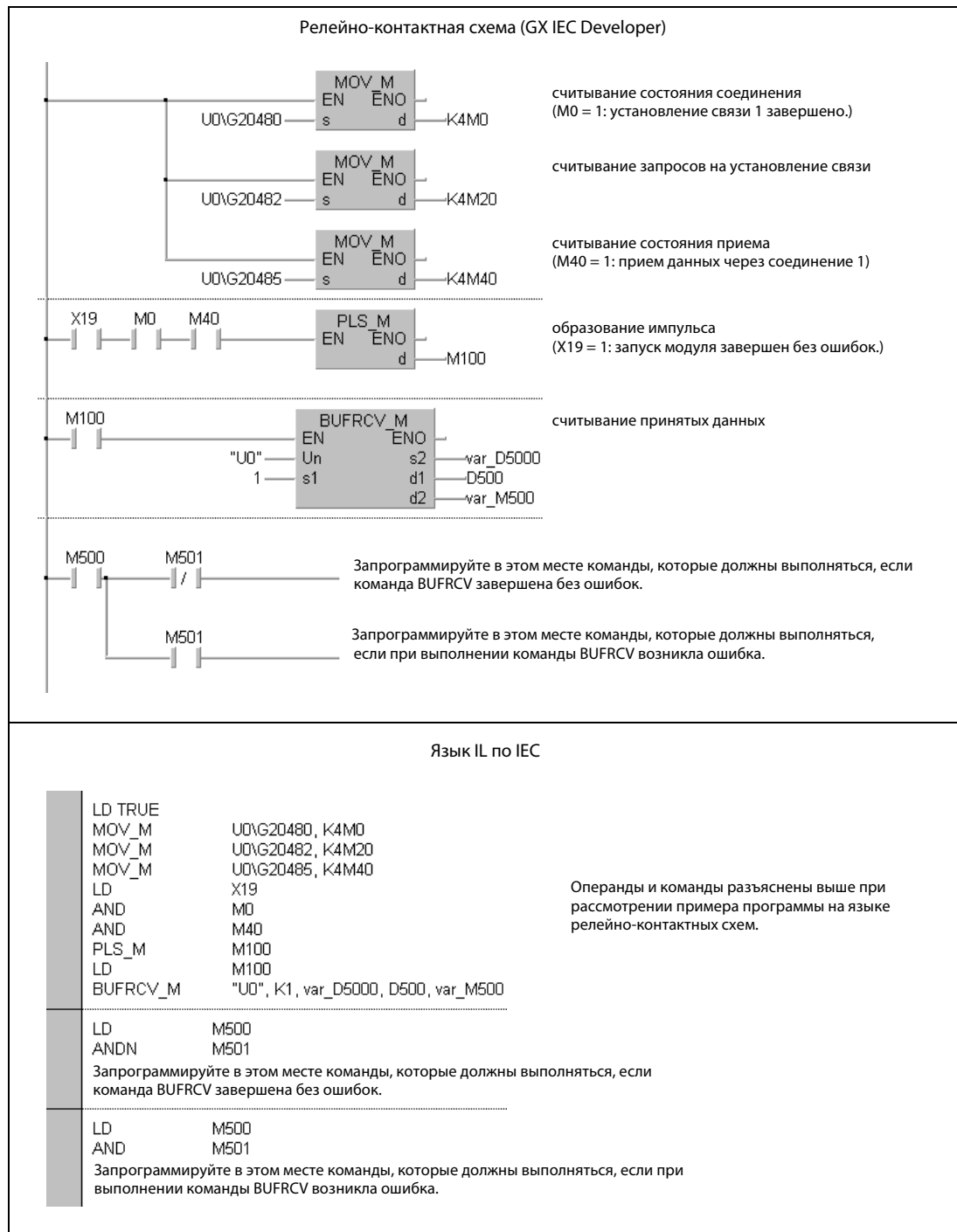
- При коде ошибки до 4FFF<sub>H</sub> указания по устранению ошибки имеются в этом руководстве (глава 13).
- При коде ошибки, начиная с C001<sub>H</sub>, более подробная информация содержится в руководствах по модулям Ethernet для серии "System Q".

**Пример**

**BUFRCV**

Следующая программа считывает данные из фиксированных буферов, принятые через соединение 1. Модуль Ethernet занимает в контроллере диапазон адресов от X/Y0 до X/Y1F.

- Редакторы по стандарту IEC (на следующей странице этот пример воспроизведен на языке списка инструкций MELSEC и языке LD в среде GX Developer)

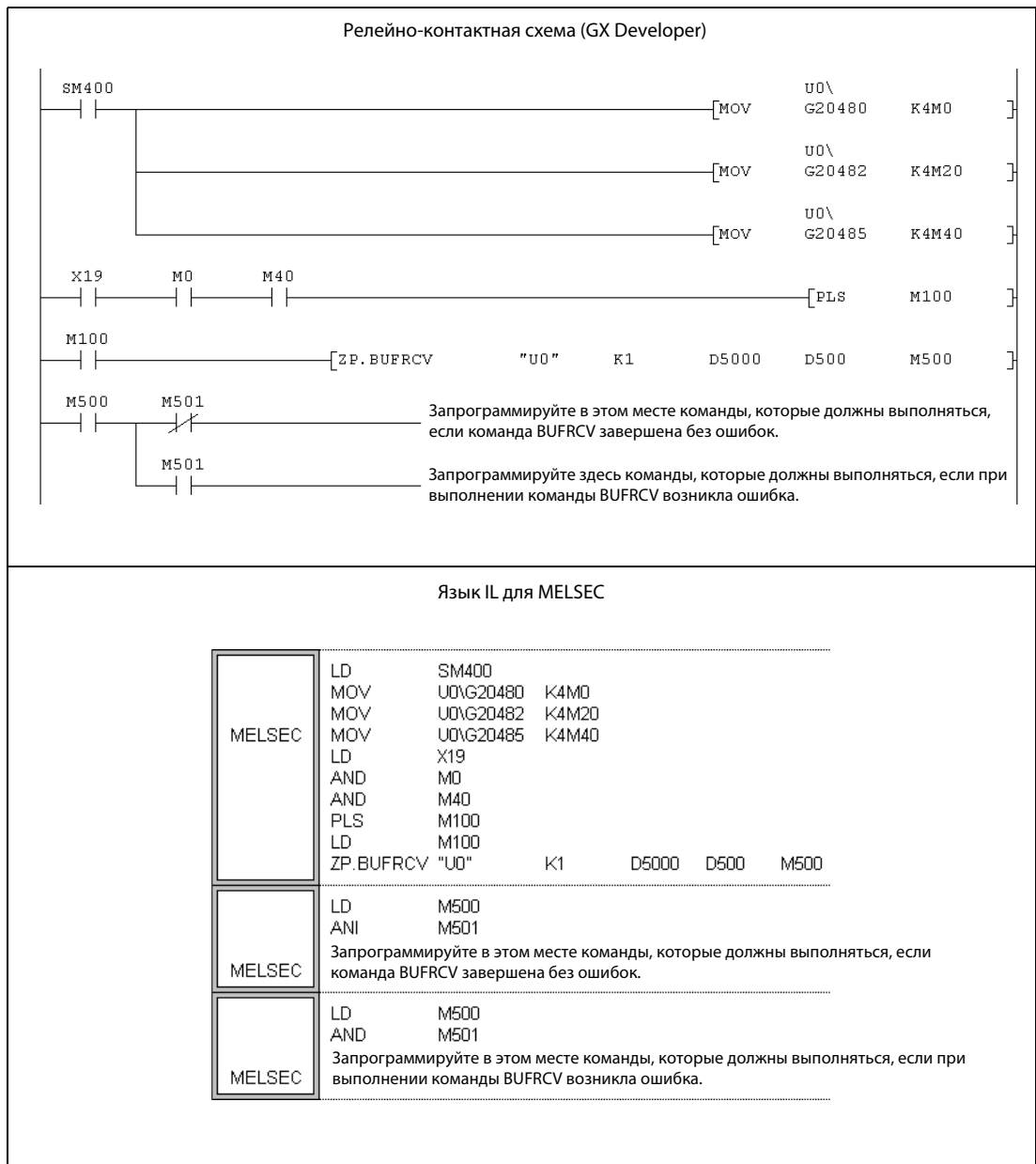


**ПРИМЕЧАНИЕ**

В редакторах по стандарту IEC среды GX IEC Developer переменные должны быть объявлены в заголовке программного компонента (POU). Если переменные не были объявлены, то при проверке или компиляции программы возникает сообщение об ошибке.

Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

- Список инструкций MELSEC и релейно-контактная схема в среде GX Developer  
Операнды и команды разъяснены на предыдущей странице при рассмотрении примера программы на языке LD в среде GX IEC Developer.



### 11.3.2 BUFRCVS

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

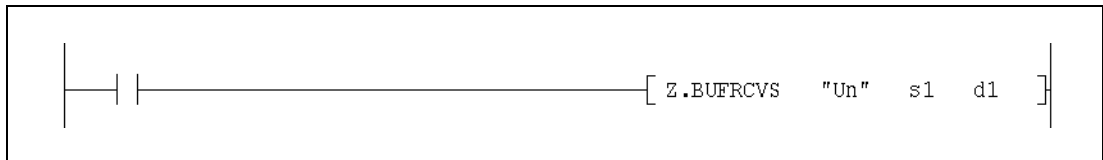
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.в.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	●	—	SM0	
d1	—	●	●	—	—	—	—	—	—		

**GX IEC Developer**



**GX Developer**



**Переменные**

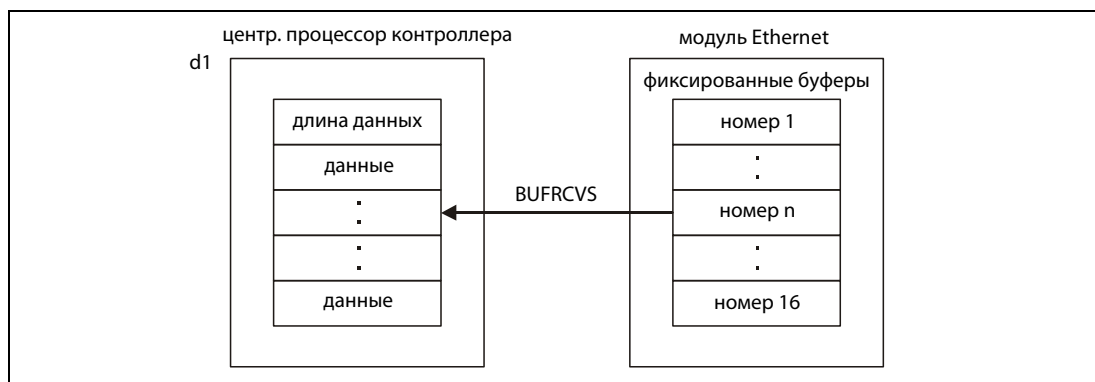
Операнд	Значение	Диапазон	Устанавливает	Тип данных
„Un“	Головной адрес модуля Ethernet на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде "U10".)	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит
s1	Номер соединения	от 1 до 16		
d1	Первый операнд области, в которой сохраняются принятые данные.			
	Операнд	Значение	Описание	Диапазон
	(d1) + 0	Длина принятых данных (количество данных, считанных из фиксированного буфера)	С процедурой (двоичные данные):	от 1 до 1017 слов
			С процедурой (данные в формате ASCII):	от 1 до 508 слов
Без процедуры (двоичные данные):			от 1 до 2016 байтов	
от (d1) + 1 до (d1) + n	Принятые данные	В этой области данные, считанные из фиксированного буфера, поочередно записываются в возрастающей последовательности.	—	система

**Принцип действия**

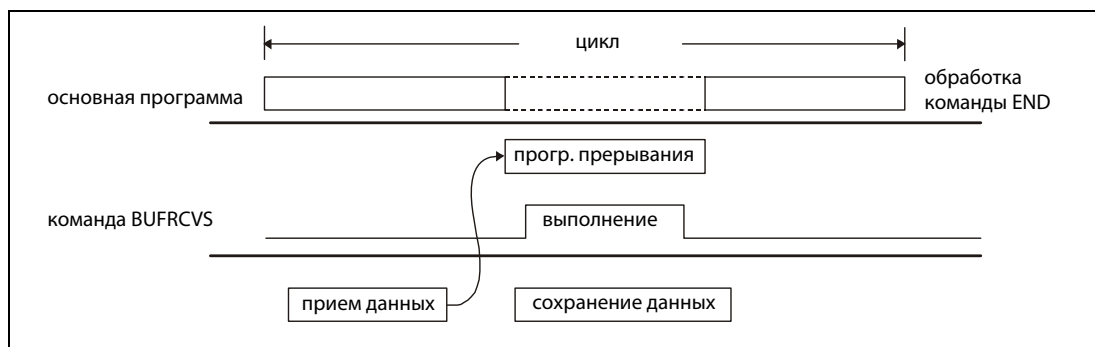
**Считывание данных из фиксированного буфера (команда вызывается в программе прерывания)**

**BUFRCVS      Считывание данных**

Команда BUFRCVS считывает из модуля Ethernet данные, которые при коммуникации с использованием фиксированного буфера были переданы из внешней станции в модуль Ethernet, и сохраняет их в центральном процессоре контроллера. В отличие от команды BUFRCV, команда BUFRCVS применяется в программе прерывания. С помощью d1 указывается, где следует сохранить данные:



Передача данных завершается в том же цикле. На рисунке ниже показана диаграмма изменения сигналов при выполнении команды BUFRCVS:



Для считывания принимаемых данных с помощью программы прерывания необходимо, чтобы в GX (IEC) Developer были запараметрированы настройки прерываний и указатели прерываний.

**ПРИМЕЧАНИЯ**

Принятые данные одного и того же соединения невозможно одновременно считывать с помощью команд BUFRCV (для основных программ) и BUFRCVS (для программ прерывания).

Команду BUFRCVS можно применять и в отношении модуля интерфейса QJ71C24.

**Источники ошибок**

Если при выполнении команды BUFRCVS возникла ошибка, то устанавливается флаг ошибки SMO и в регистр SD0 записывается код ошибки. Коды ошибок подробно разъяснены в следующих руководствах:

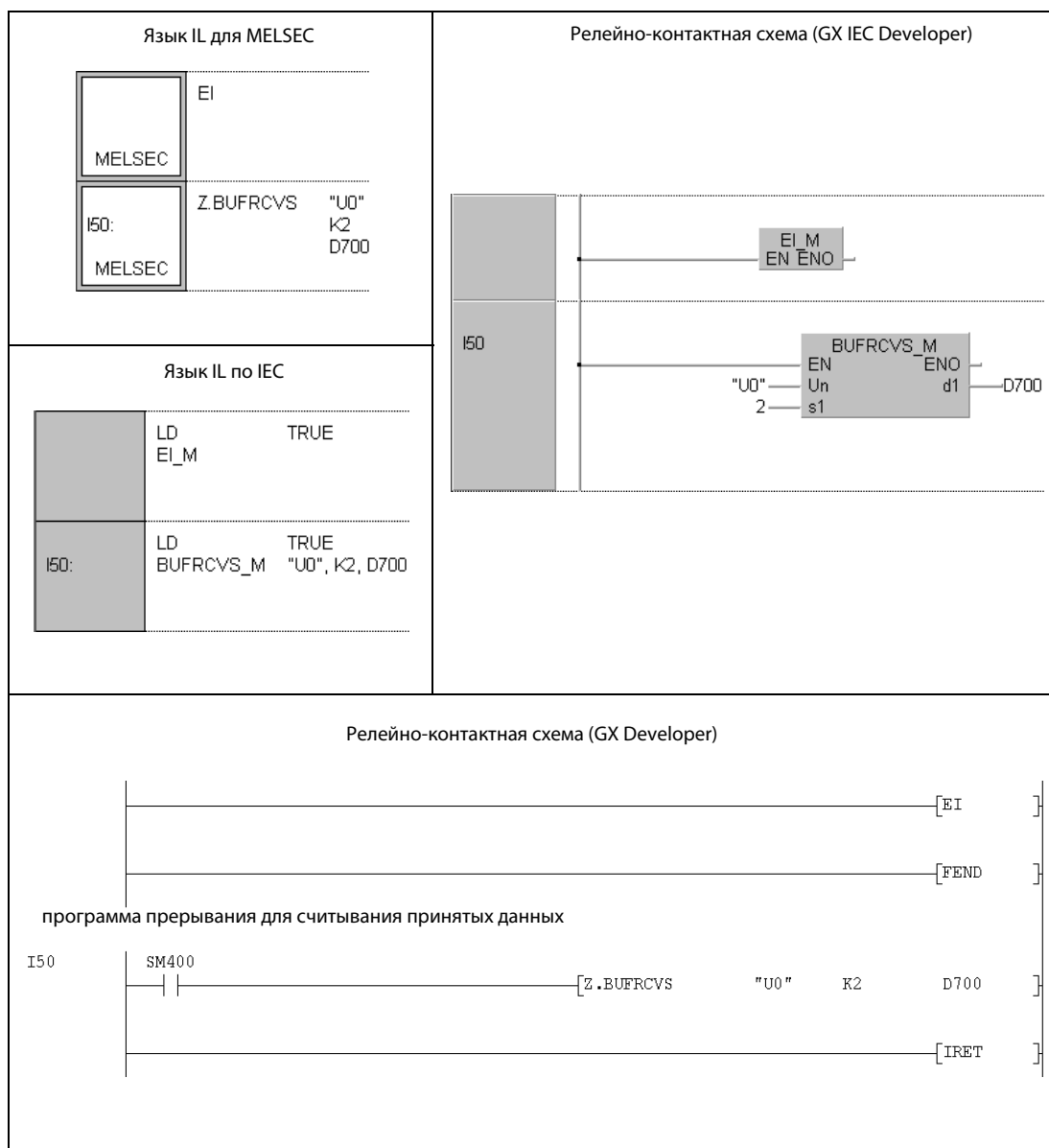
- При коде ошибки до 4FFF<sub>H</sub> указания по устранению ошибки имеются в этом руководстве (глава 13).
- При коде ошибки, начиная с C001<sub>H</sub>, более подробная информация содержится в руководствах по модулям Ethernet для серии "System Q".



**Пример**

**BUFRCVS**

Следующая программа считывает данные из фиксированных буферов модуля Ethernet с головным адресом X/Y0, принятые через соединение 2.



### 11.3.3 BUFSND

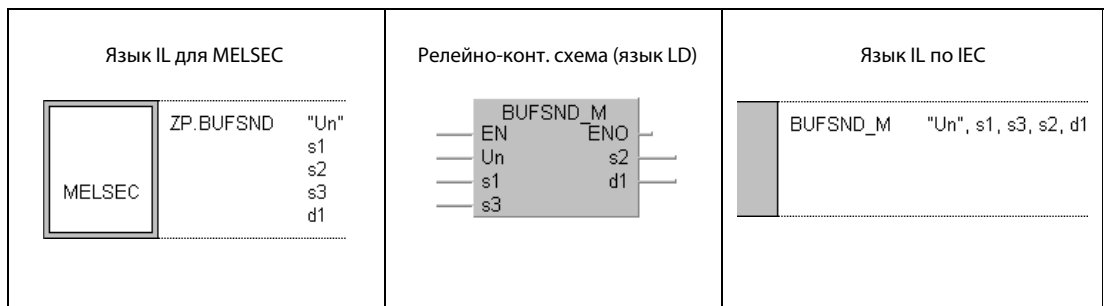
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

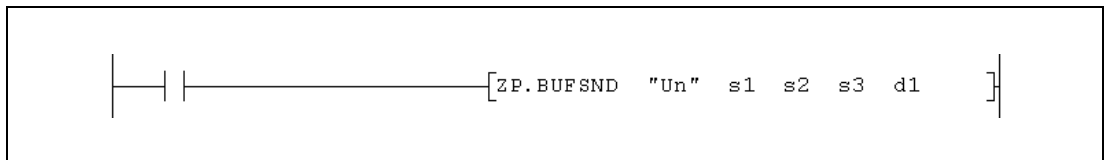
Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.в.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	●	—		
s2	—	●	●	—	—	—	—	—	—		
s3	—	●	●	—	—	—	—	—	—		
d1	●	●	●	—	—	—	—	—	—		

GX IEC  
Developer



GX  
Developer



## Переменные

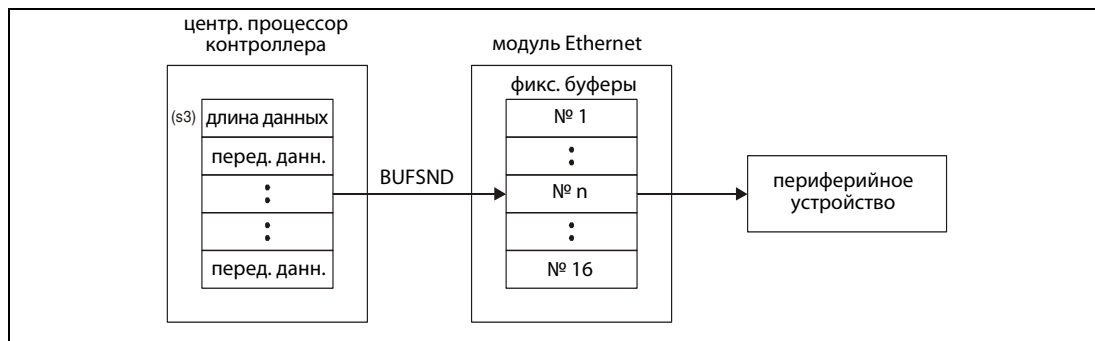
Операнд	Значение	Диапазон	Устанавливает	Тип данных	
„Un“	Головной адрес модуля Ethernet на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде "U10".)	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит	
s1	Номер соединения	от 1 до 16			
s2	Первый операнд области, содержащей информацию для выполнения команды.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(s2) + 0	Системная область	Используется системой	—	система
(s2) + 1	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды: 0000 <sub>H</sub> : безошибочная обработка Любое иное значение кроме 0000 <sub>H</sub> : При обработке возникла ошибка. Записанное значение представляет собой код ошибки, который разъяснен либо в руководстве по модулю Ethernet, либо в глава 13 этого руководства.			
s3	Первый операнд области, в которой сохранены передаваемые данные.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(s3) + 0	Длина передаваемых данных	Указание количества данных, которое при передаче с процедурой (двоичные данные) требуется ввести в фиксированный буфер.	от 1 до 1017 слов	пользователь
			Указание количества данных, которое при передаче с процедурой (данные в формате ASCII) требуется ввести в фиксированный буфер.	от 1 до 508 слов	
Указание количества данных, которое при передаче без процедуры (двоичные данные) требуется ввести в фиксированный буфер.			от 1 до 2046 байтов		
от (s3) + 1 до (s3) + n	Передаваемые данные	Данные, передаваемые в модуль Ethernet	—		
d1	Битовый операнд, устанавливаемый на один цикл после выполнения команды BUFSND. С помощью (d1) + 1 сигнализируется ошибочное завершение.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(d1) + 0	Команда выполнена	Показывает завершение команды BUFSND. ВКЛ. : команда выполнена ВыКЛ. : команда не выполнена	—	система
(d1) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды BUFSND. ВКЛ.: команда выполнена с ошибкой ВыКЛ.: команда выполнена без ошибок	—		

**Принцип действия**

**Запись данных в фиксированные буферы**

**BUFSND Запись данных**

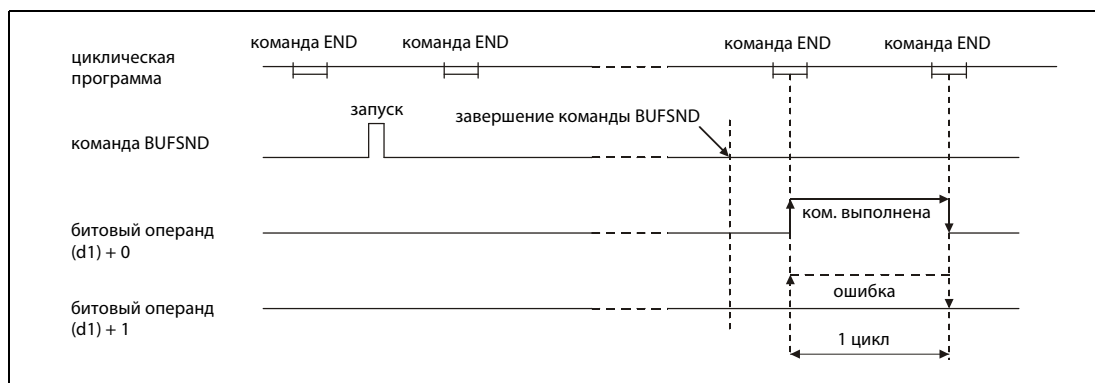
Команда BUFSND передает в модуль Ethernet данные, которые при коммуникации с использованием фиксированного буфера требуется передать на внешнюю станцию, соединенную с модулем Ethernet. Данные хранятся в памяти центрального процессора, начиная с операнда, указанного в (s3) + 1:



Завершено ли выполнение команды BUFSND, можно определить по состоянию битовых операндов (d1) + 0 и (d1) + 1:

- Битовый операнд (d1) + 0 устанавливается при выполнении команды END того цикла, в котором команда BUFSND была завершена. При обработке следующей команды END указанный в (d1) + 0 битовый операнд снова сбрасывается.
- Битовый операнд (d1) + 1 показывает ошибку при выполнении команды BUFSND. При безошибочном выполнении этот битовый операнд остается сброшенным. В случае ошибки (d1) + 1 устанавливается при выполнении команды END того цикла, в котором команда BUFSND была завершена. При обработке следующей команды END битовый операнд, указанный в (d1) + 1, снова сбрасывается.

На рисунке ниже показана диаграмма изменения сигналов при выполнении команды BUFSND:



Команда BUFSND выполняется при выполненном входном условии команды.

**Источники ошибок**

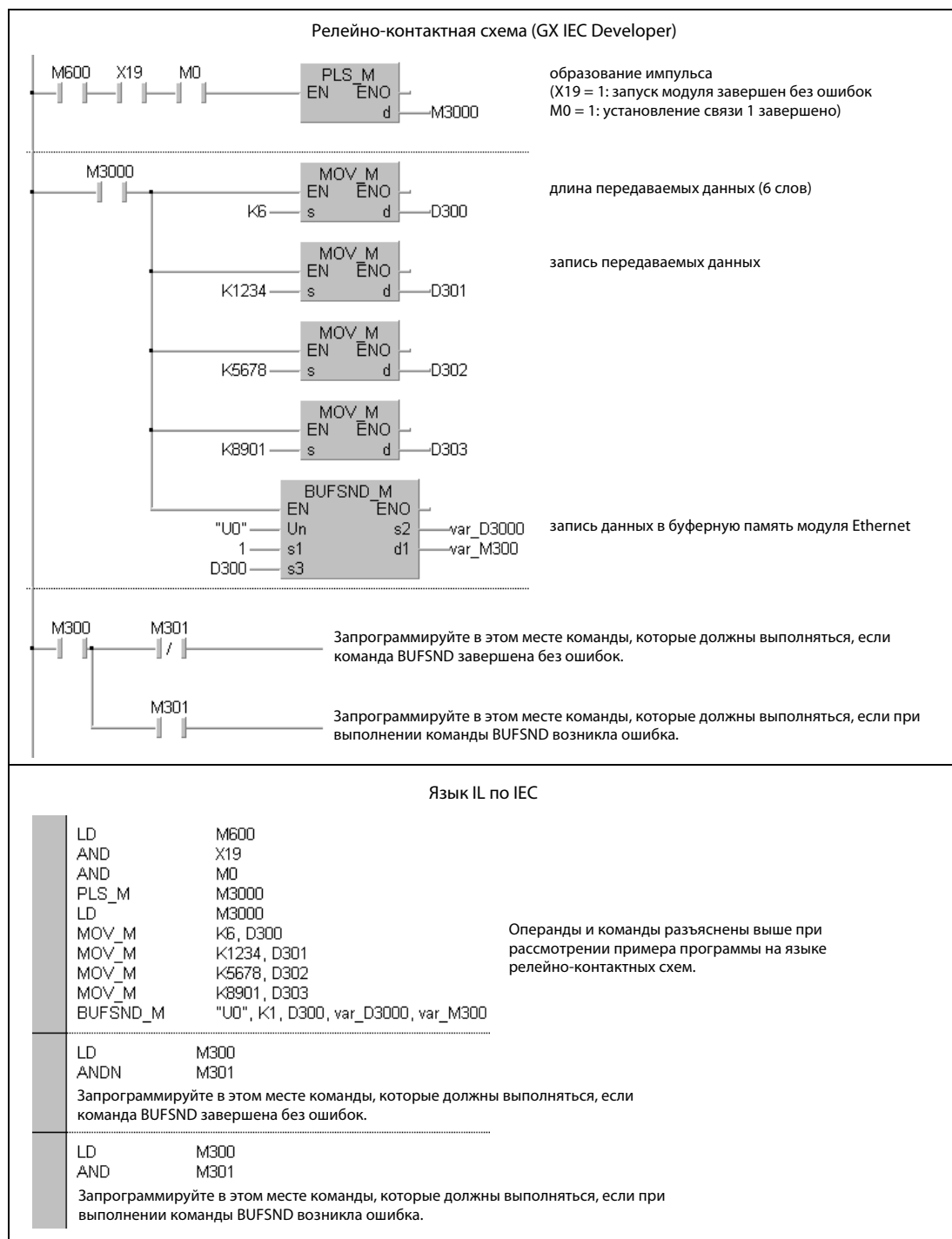
Если команда BUSND выполнена с ошибкой, то устанавливается операнд (d1) + 1 и соответствующий код ошибки записывается в операнд (s2) + 1. Подробные сведения об отдельных кодах ошибок имеются в следующих руководствах:

- При коде ошибки до 4FFF<sub>H</sub> указания по устранению ошибки имеются в этом руководстве (глава 13).
- При коде ошибки, начиная с C001<sub>H</sub>, более подробная информация содержится в руководствах по модулям Ethernet для серии "System Q".

**Пример** BUFSND

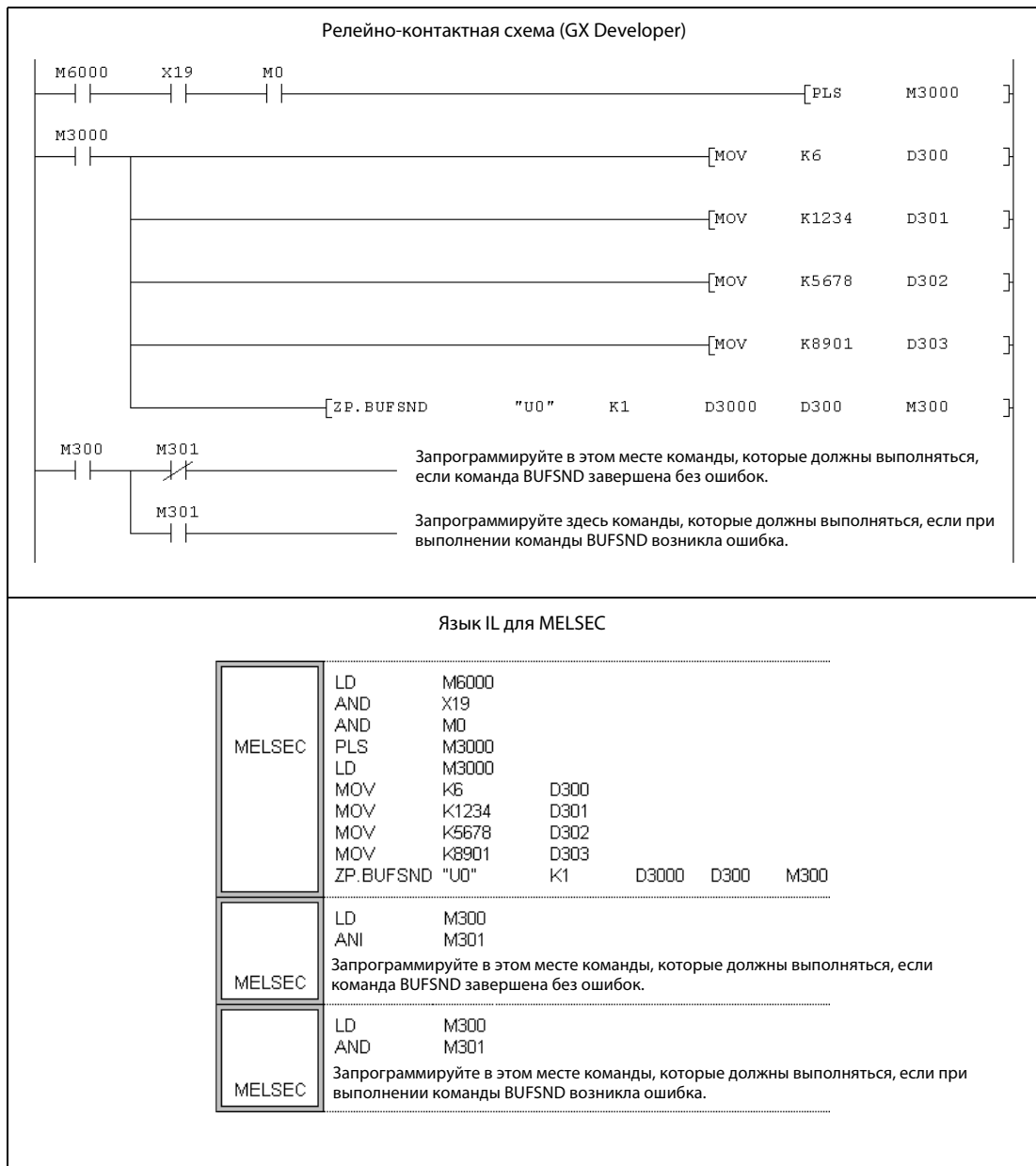
Следующая программа записывает данные в фиксированный буфер связи 1. Модуль Ethernet имеет головной адрес X/Y0.

- Редакторы по стандарту IEC (на следующей странице этот пример воспроизведен на языке списка инструкций MELSEC и языке LD в среде GX Developer)



**ПРИМЕЧАНИЕ** В редакторах по стандарту IEC среды GX IEC Developer переменные должны быть объявлены в заголовке программного компонента (POU). Если переменные не были объявлены, то при проверке или компиляции программы возникает сообщение об ошибке. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

- Список инструкций MELSEC и релейно-контактная схема в среде GX Developer  
Операнды и команды разъяснены на предыдущей странице при рассмотрении примера программы на языке LD в среде GX IEC Developer.



### 11.3.4 OPEN

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

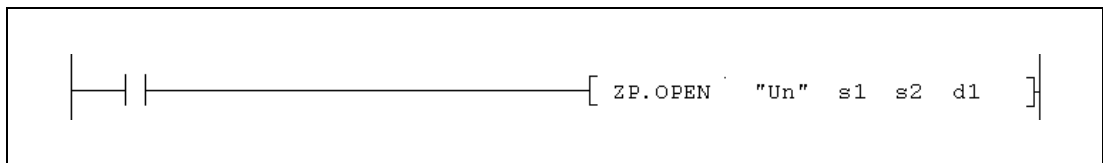
Операнды MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользov.)		Регистры файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	●	—		
s2	—	●	●	—	—	—	—	—	—		
d1	●	●	●	—	—	—	—	—	—		

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных
„Un“	Головной адрес модуля Ethernet на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде "U10".)	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит
s1	Номер соединения	от 1 до 16		

## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных
Первый операнд области, содержащей информацию для выполнения команды.				
Операнд	Значение	Описание	Диапазон	Устанавливает
(s2) + 0	Источник параметров	<p>Указание, какие параметры должны использоваться для установления связи.</p> <ul style="list-style-type: none"> <li>0000<sub>H</sub>: связь устанавливается на основе параметров, указанных в GX (IEC) Developer</li> <li>8000<sub>H</sub>: связь устанавливается на основе параметров, сохраненных в операндах от (s2) + 2 до (s2) + 9.</li> </ul>	0000 <sub>H</sub> или 8000 <sub>H</sub>	пользователь
(s2) + 1	Состояние выполнения команды	<p>Показывает, возникла ли ошибка при обработке команды.</p> <p>0000<sub>H</sub>: безошибочная обработка</p> <p>Любое иное значение кроме 0000<sub>H</sub>: При обработке возникла ошибка. Записанное значение представляет собой код ошибки, разъяснение которого можно найти либо в руководстве по модулю Ethernet, либо в глава 13 этого руководства.</p>	—	система
s2		<p>Отдельные биты этого операнда служат для установки свойств соединения.</p> <ul style="list-style-type: none"> <li>Бит 0: применение фиксированных буферов 0: буфер служит в качестве буфера передачи, передача фиксированных буферов отключена 1: буфер служит для приема данных</li> <li>Бит 1: проверка, существует ли партнерская станция 0: без проверки существования 1: проверка выполняется</li> <li>Бит 7: попарное открытие связи 0: без попарной связи 1: попарная связь</li> <li>Бит 8: выбор протокола передачи 0: TCP/IP 1: UDP/IP</li> <li>Бит 9: процедура при передаче фиксированных буферов 0: с процедурой 1: без процедуры</li> <li>Бит 13: активация прерывания при приеме фиксированных буферов 0: без прерывания 1: прерывание активируется</li> <li>Биты 14 и 15: активное или пассивное установление связи бит 15/14 = 00: активное открытие связи или UDP/IP бит 15/14 = 10: связь открывается "непассивно" бит 15/14 = 11: связь открывается "полностью пассивно"</li> </ul>	Эти значения образуются путем установки отдельных битов.	пользователь
				BIN, 16 бит



## Переменные

Операнд	Значение			Диапазон	Устанавливает	Тип данных
	Операнд	Значение	Описание			
s2	(s2) + 3	Номер порта модуля Ethernet	Укажите здесь номер порта модуля Ethernet.	408 <sub>H</sub> ... 1388 <sub>H</sub> 138B <sub>H</sub> ... FFFE <sub>H</sub>	пользователь	BIN, 16 бит
	(s2) + 4 (s2) + 5	IP-адрес целевой станции	IP-адрес станции, с которой происходит коммуникация. Если указан IP-адрес FFFFFFFF <sub>H</sub> , данные передаются по широковещательному принципу.	1 <sub>H</sub> ... FFFFFFF <sub>H</sub>		
	(s2) + 6	Номер порта целевой станции	Номер порта партнерской станции, с которой происходит коммуникация. (FFFF <sub>H</sub> = широковещание)	401 <sub>H</sub> ... FFFF <sub>H</sub>		
	от (s2) + 7 до (s2) + 9	Адрес целевой станции Ethernet	Если партнерская станция поддерживает функцию ARP, укажите либо 000000000000 <sub>H</sub> , либо FFFFFFFF <sub>H</sub> . Если партнерская станция не поддерживает функцию ARP, укажите индивидуальный адрес Ethernet.	см. разъяснение рядом		
d1	Битовый операнд, устанавливаемый на один цикл после выполнения команды OPEN. С помощью следующего операнда (d1) + 1 сигнализируется ошибочное завершение.					бит
	Операнд	Значение	Описание	Диапазон	Устанавливает	
	(d1) + 0	Команда выполнена	Показывает завершение команды OPEN. ВКЛ.: команда выполнена Выкл.: команда не выполнена	—	система	
(d1) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды OPEN. ВКЛ.: команда выполнена с ошибкой Выкл.: команда выполнена без ошибок	—			

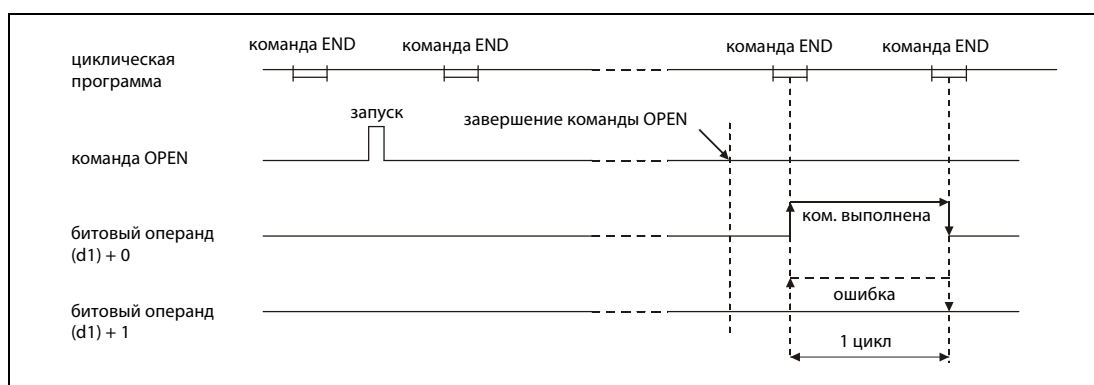
**Принцип действия****Установление связи****OPEN Открытие связи**

С помощью этой команды можно открыть (установить) указанное в  $s1$  соединение модуля Ethernet, имеющего головной адрес  $Un$ .

Выполнение команды OPEN можно проверить по состоянию битовых операндов  $(d1) + 0$  и  $(d1) + 1$ :

- Битовый операнд  $(d1) + 0$  устанавливается при выполнении команды END того цикла, в котором команда OPEN была завершена. При обработке следующей команды END этот битовый операнд снова сбрасывается.
- Битовый операнд  $(d1) + 1$  показывает ошибку при выполнении команды OPEN. При безошибочном выполнении этот битовый операнд остается сброшенным. В случае ошибки  $(d1) + 1$  устанавливается при выполнении команды END того цикла, в котором команда OPEN была завершена. При обработке следующей команды END битовый операнд, указанный в  $(d1) + 1$ , снова сбрасывается.

На рисунке ниже показана диаграмма изменения сигналов при выполнении команды OPEN:



Команда OPEN выполняется при выполненном входном условии команды.

**ПРИМЕЧАНИЕ**

*Не открывайте и не закрывайте одно и то же соединение одновременно с помощью входных и выходных сигналов и команд OPEN или CLOSE. Это может привести к неправильному функционированию модуля.*

**Источники ошибок**

Если команда OPEN выполнена с ошибкой, то устанавливается операнд  $(d1) + 1$  и в операнд  $(s2) + 1$  записывается код ошибки. Коды ошибок подробно разъяснены в следующих руководствах:

- При коде ошибки до  $4FFF_H$  указания по устранению ошибки имеются в этом руководстве (глава 13).
- При коде ошибки, начиная с  $C001_H$ , более подробная информация содержится в руководствах по модулям Ethernet для серии "System Q".

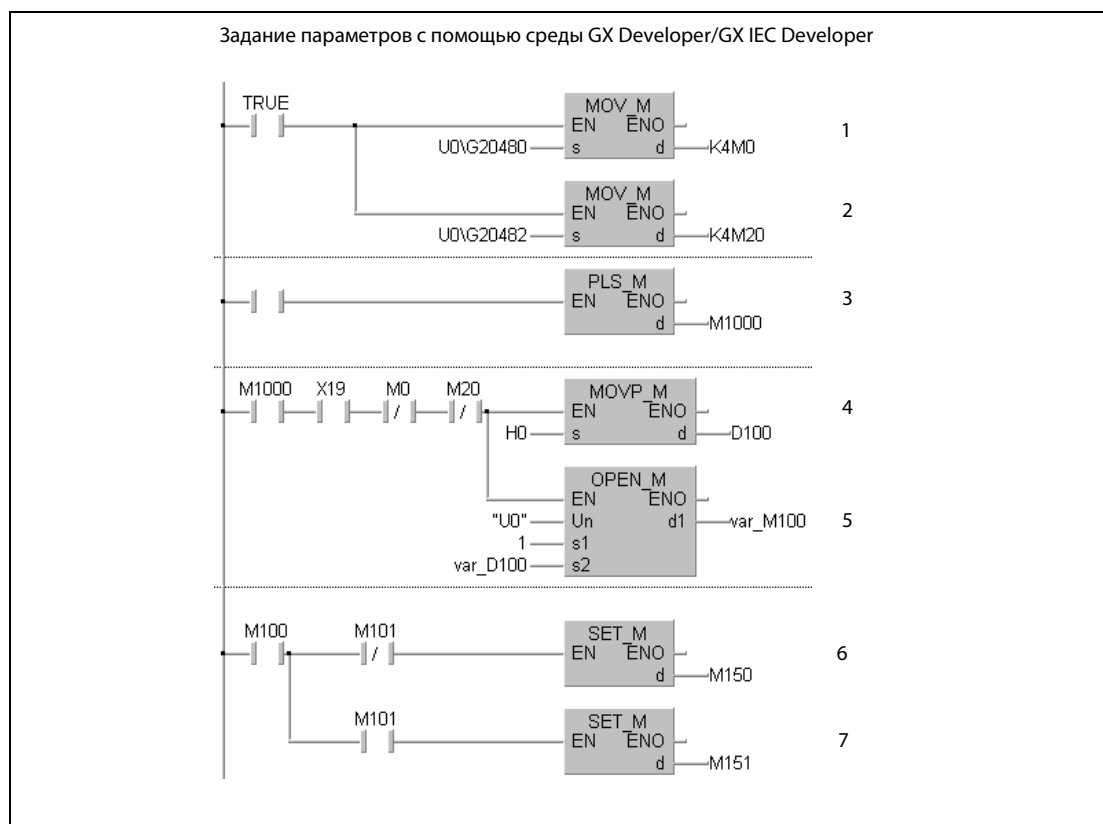
**Пример** OPEN

Следующая программа открывает соединение 1 модуля Ethernet, имеющего головной адрес X/Y0, для коммуникации по протоколу TCP/IP.

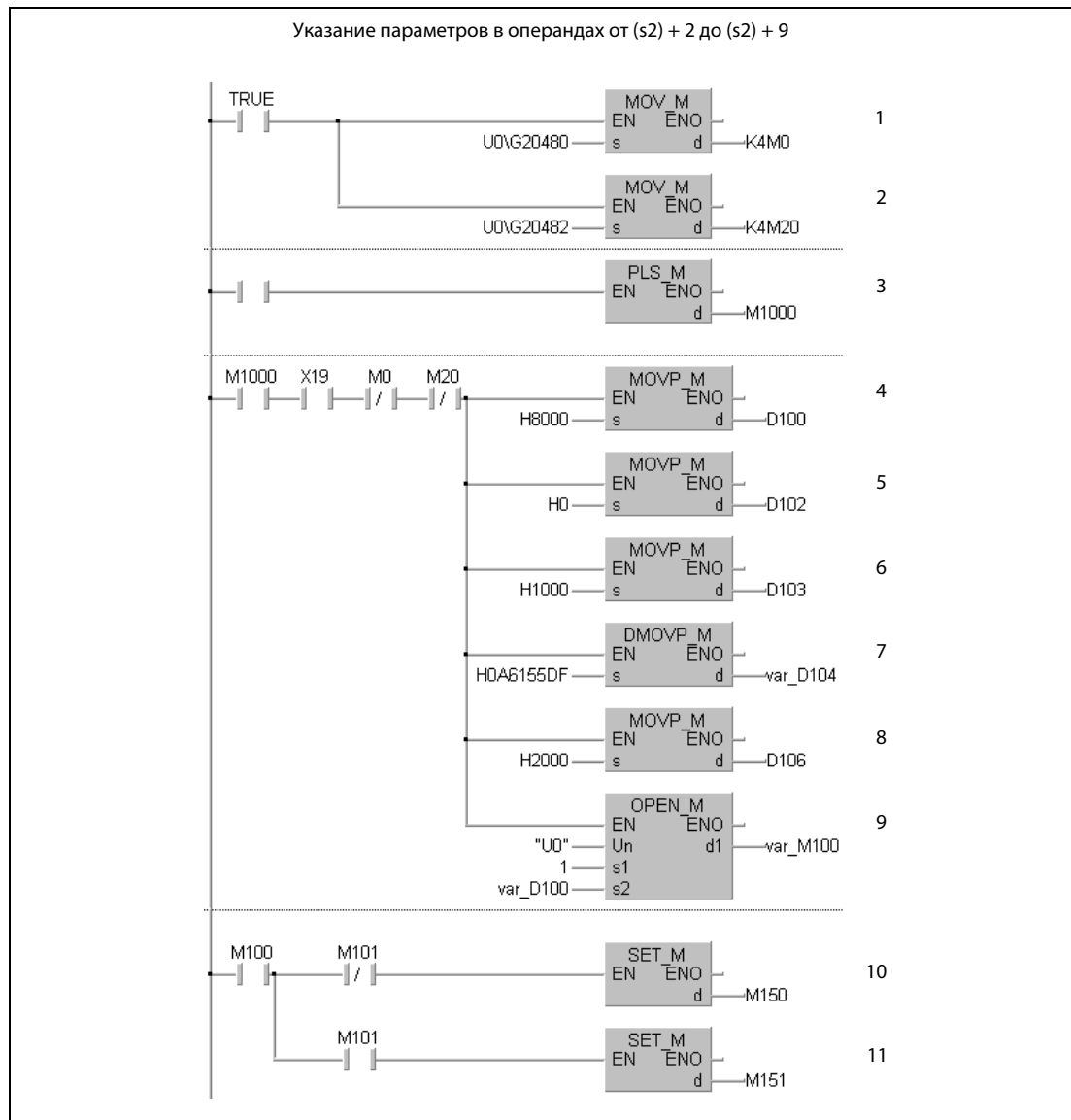
**ПРИМЕЧАНИЕ** В редакторах по стандарту IEC среды GX IEC Developer переменные должны быть объявлены в заголовке программного компонента (POU). Если переменные не были объявлены, то при проверке или компилировании программы возникает сообщение об ошибке. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

- Релейно-контактная схема (GX IEC Developer)

В следующем примере необходимо предварительно настроить параметры с помощью среды программирования. Пример передачи параметров с помощью команды OPEN имеется на следующей странице.



- 1 считывание состояния соединения (M0 = 1: установление связи 1 завершено)
- 2 считывание запросов на установление связи (M20 = 1: затребовано установление связи 1)
- 3 из сигнала открытия сеанса связи формируется импульс
- 4 указание источника параметров (0000<sub>H</sub> = внешнее задание)
- 5 открытие связи 1
- 6 Если связь открыта без ошибок, устанавливается маркер M150.
- 7 Если при открытии связи возникла ошибка, устанавливается маркер M151.



- 1 считывание состояния соединения (M0 = 1: установление связи 1 завершено)
- 2 считывание запросов на установление связи (M20 = 1: затребовано установление связи 1)
- 3 Из сигнала открытия сеанса связи формируется импульс.
- 4 указание источника параметров (8000<sub>H</sub> = параметры в операндах от (s2) + 2 до (s2) + 9)
- 5 указание типа соединения в (s2) + 2
- 6 указание номера порта модуля Ethernet в (s2) + 3
- 7 указание IP-адреса (10.97.85.223) внешней станции в (s2) + 4 и (s2) + 5
- 8 указание номера порта внешней станции в (s2) + 6
- 9 открытие связи 1
- 10 Если связь открыта без ошибок, устанавливается маркер M150.
- 11 Если при открытии связи возникла ошибка, устанавливается маркер M151.

## ● Язык IL по стандарту IEC

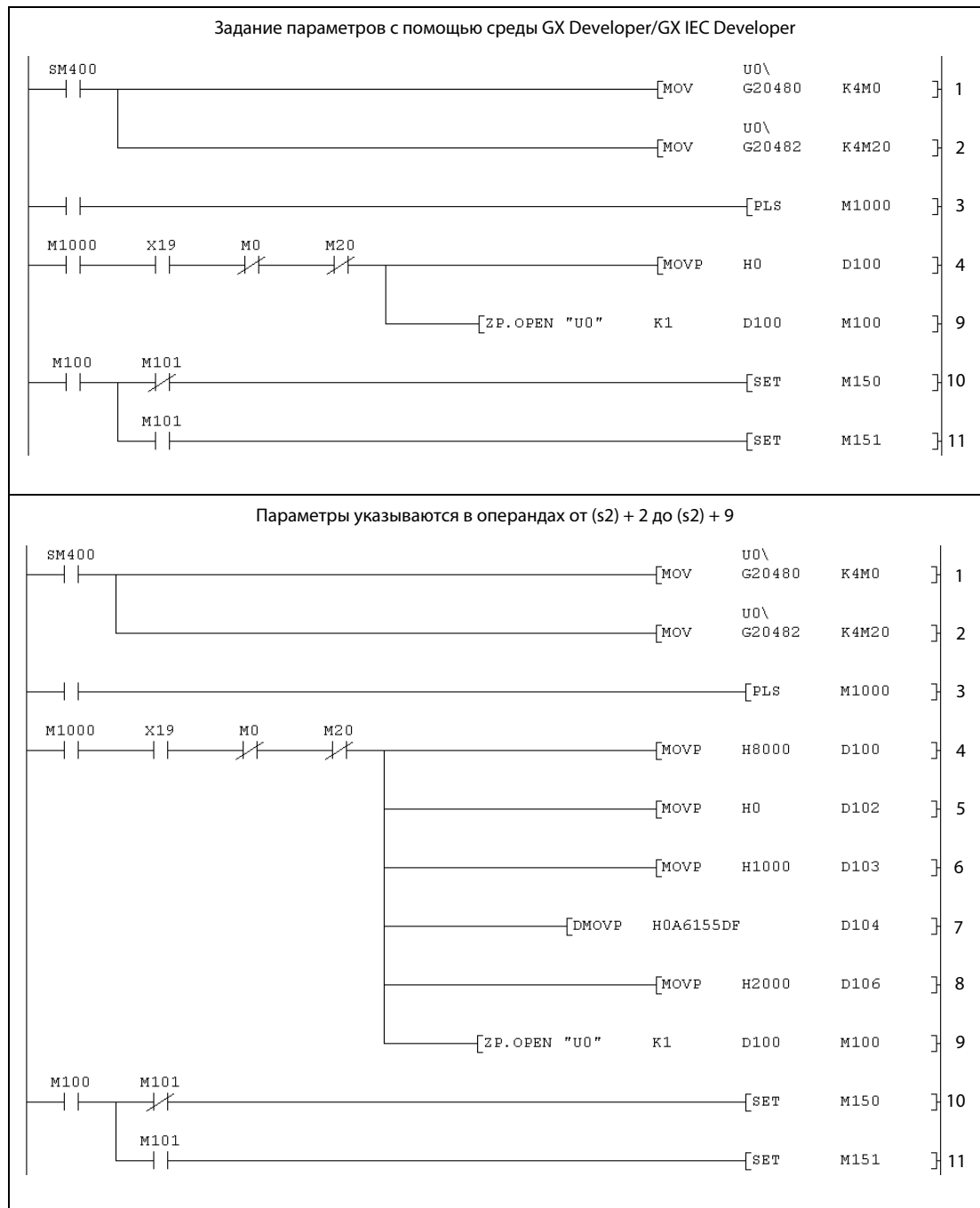
Задание параметров с помощью среды GX Developer/GX IEC Developer				
LD	TRUE			
MOV_M	U0\G20480, K4M0	_____		1
MOV_M	U0\G20482, K4M20	_____		2
LD	M123			
PLS_M	M1000	_____		3
LD	M1000			
AND	X19			
ANDN	M0			
ANDN	M20			
MOVP_M	H0, D100	_____		4
OPEN_M	"U0", K1, var_D100, var_M100	_____		9
LD	M100			
ANDN	M101			
SET_M	M150	_____		10
LD	M100			
AND	M101			
SET_M	M151	_____		11

Указание параметров в операндах от (s2) + 2 до (s2) + 9				
LD	TRUE			
MOV_M	U0\G20480, K4M0	_____		1
MOV_M	U0\G20482, K4M20	_____		2
LD	M123			
PLS_M	M1000	_____		3
LD	M1000			
AND	X19			
ANDN	M0			
ANDN	M20			
MOVP_M	H8000, D100	_____		4
MOVP_M	H0, D102	_____		5
MOVP_M	H1000, D103	_____		6
DMOVP_M	H0A6155DF, var_D104	_____		7
MOVP_M	H2000, D106	_____		8
OPEN_M	"U0", K1, var_D100, var_M100	_____		9
LD	M100			
ANDN	M101			
SET_M	M150	_____		10
LD	M100			
AND	M101			
SET_M	M151	_____		11

- 1 считывание состояния соединения (M0 = 1: установление связи 1 завершено)
- 2 считывание запросов на установление связи (M20 = 1: затребовано установление связи 1)
- 3 Из сигнала открытия сеанса связи формируется импульс.
- 4 указание источника параметров (0000<sub>H</sub> = внешний, 8000<sub>H</sub> = операнды от (s2) + 2 до (s2) + 9)
- 5 указание типа соединения в (s2) + 2
- 6 указание номера порта модуля Ethernet в (s2) + 3
- 7 указание IP-адреса (10.97.85.223) внешней станции в (s2) + 4 и (s2) + 5
- 8 указание номера порта внешней станции в (s2) + 6
- 9 открытие связи 1
- 10 Если связь открыта без ошибок, устанавливается маркер M150.
- 11 Если при открытии связи возникла ошибка, устанавливается маркер M151.

● Релейно-контактная схема (GX Developer)



- 1 считывание состояния соединения (M0 = 1: установление связи 1 завершено)
- 2 считывание запросов на установление связи (M20 = 1: затребовано установление связи 1)
- 3 Из сигнала открытия сеанса связи формируется импульс.
- 4 указание источника параметров (0000<sub>H</sub> = внешний, 8000<sub>H</sub> = операнды от (s2) + 2 до (s2) + 9))
- 5 указание типа соединения в (s2) + 2
- 6 указание номера порта модуля Ethernet в (s2) + 3
- 7 указание IP-адреса (10.97.85.223) внешней станции в (s2) + 4 и (s2) + 5
- 8 указание номера порта внешней станции в (s2) + 6
- 9 открытие связи 1
- 10 Если связь открыта без ошибок, устанавливается маркер M150.
- 11 Если при открытии связи возникла ошибка, устанавливается маркер M151.

## ● Язык IL для MELSEC

Задание параметров с помощью среды GX Developer/GX IEC Developer					
MELSEC	LD	SM400			
	MOV	U0VG20480	K4M0	_____	1
	MOV	U0VG20482	K4M20	_____	2
	LD	M123			
	PLS	M1000		_____	3
	LD	M1000			
	AND	X19			
	ANI	M0			
	ANI	M20			
MOV	H0	D100	_____	4	
ZP.OPEN	"U0"	K1	D100 M100	_____	9
MELSEC	LD	M100			
	ANI	M101			
	SET	M150		_____	10
MELSEC	LD	M100			
	AND	M101			
	SET	M151		_____	11

Параметры указываются в операндах от (s2) + 2 до (s2) + 9					
MELSEC	LD	SM400			
	MOV	U0VG20480	K4M0	_____	1
	MOV	U0VG20482	K4M20	_____	2
	LD	M123			
	PLS	M1000		_____	3
	LD	M1000			
	AND	X19			
	ANI	M0			
	ANI	M20			
MOV	H8000	D100	_____	4	
MOV	H0	D102	_____	5	
MOV	H1000	D103	_____	6	
DMOV	HA6155DF	D104	_____	7	
MOV	H2000	D106	_____	8	
ZP.OPEN	"U0"	K1	D100 M100	_____	9
MELSEC	LD	M100			
	ANI	M101			
	SET	M150		_____	10
MELSEC	LD	M100			
	AND	M101			
	SET	M151		_____	11

- 1 считывание состояния соединения (M0 = 1: установление связи 1 завершено)
- 2 считывание запросов на установление связи (M20 = 1: затребовано установление связи 1)
- 3 Из сигнала открытия сеанса связи формируется импульс.
- 4 указание источника параметров (0000<sub>H</sub> = внешний, 8000<sub>H</sub> = операнды от (s2) + 2 до (s2) + 9)
- 5 указание типа соединения в (s2) + 2
- 6 указание номера порта модуля Ethernet в (s2) + 3
- 7 указание IP-адреса (10.97.85.223) внешней станции в (s2) + 4 и (s2) + 5
- 8 указание номера порта внешней станции в (s2) + 6
- 9 открытие связи 1
- 10 Если связь открыта без ошибок, устанавливается маркер M150.
- 11 Если при открытии связи возникла ошибка, устанавливается маркер M151.

11.3.5 CLOSE

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

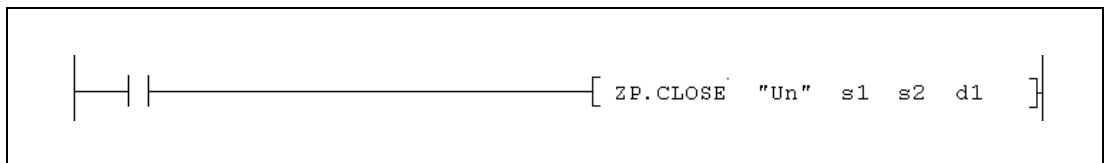
Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	●	—		
s2	—	●	●	—	—	—	—	—	—		
d1	●	●	●	—	—	—	—	—	—		

GX IEC  
Developer

<p>Язык IL для MELSEC</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p>MELSEC</p> </div> <p>ZP.CLOSE "Un" s1 s2 d1</p>	<p>Релейно-конт. схема (язык LD)</p>	<p>Язык IL по IEC</p> <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 10px auto;"> <p>CLOSE_M "Un", s1, s2, d1</p> </div>
--	--------------------------------------	--

GX  
Developer





## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных	
„Un“	Головной адрес модуля Ethernet на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде "U10".)	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит	
s1	Номер соединения	от 1 до 16			
s2	Первый операнд области, содержащей информацию для выполнения команды				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(s2) + 0	системная область	Используется системой	—	система
(s2) + 1	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды: 0000 <sub>H</sub> : безошибочная обработка Любое иное значение кроме 0000 <sub>H</sub> : При обработке возникла ошибка. Записанное значение представляет собой код ошибки, который разъяснен либо в руководстве по модулю Ethernet, либо в глава 13 этого руководства.			
d1	Битовый операнд, устанавливаемый на один цикл после выполнения команды CLOSE. С помощью следующего операнда (d1) + 1 сигнализируется ошибочное завершение.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(d1) + 0	Команда выполнена	Показывает завершение команды CLOSE. Вкл.: команда выполнена Выкл.: команда не выполнена	—	система
(d1) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды CLOSE. Вкл.: команда выполнена с ошибкой Выкл.: команда выполнена без ошибок	—		

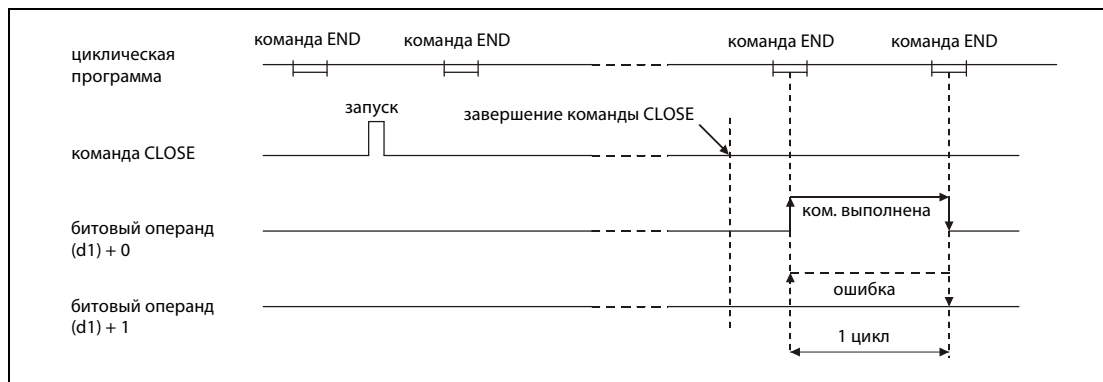
**Принцип действия****Закрытие связи****CLOSE      Закрытие связи**

С помощью этой команды можно закрыть указанное в  $s1$  соединение модуля Ethernet, имеющего головной адрес  $Un$ . Сеанс связи заканчивается.

Завершено ли выполнение команды CLOSE, можно проверить по состоянию битовых операндов  $(d1) + 0$  и  $(d1) + 1$ :

- Битовый операнд  $(d1) + 0$  устанавливается при выполнении команды END того цикла, в котором команда CLOSE была завершена. При обработке следующей команды END этот битовый операнд снова сбрасывается.
- Битовый операнд  $(d1) + 1$  показывает ошибку при выполнении команды CLOSE. При безошибочном выполнении этот битовый операнд остается сброшенным. В случае ошибки  $(d1) + 1$  устанавливается при выполнении команды END того цикла, в котором команда CLOSE была завершена. При обработке следующей команды END битовый операнд, указанный в  $(d1) + 1$ , снова сбрасывается.

На рисунке ниже показана диаграмма изменения сигналов при выполнении команды CLOSE:



Команда CLOSE выполняется при выполненном входном условии команды.

**ПРИМЕЧАНИЕ**

Не открывайте и не закрывайте одно и то же соединение одновременно с помощью входных и выходных сигналов и команды OPEN или CLOSE. Это может привести к неправильному функционированию модуля.

**Источники ошибок**

Если команда CLOSE выполнена с ошибкой, то устанавливается операнд  $(d1) + 1$  и в операнд  $(s2) + 1$  записывается код ошибки. Коды ошибок подробно разъяснены в следующих руководствах:

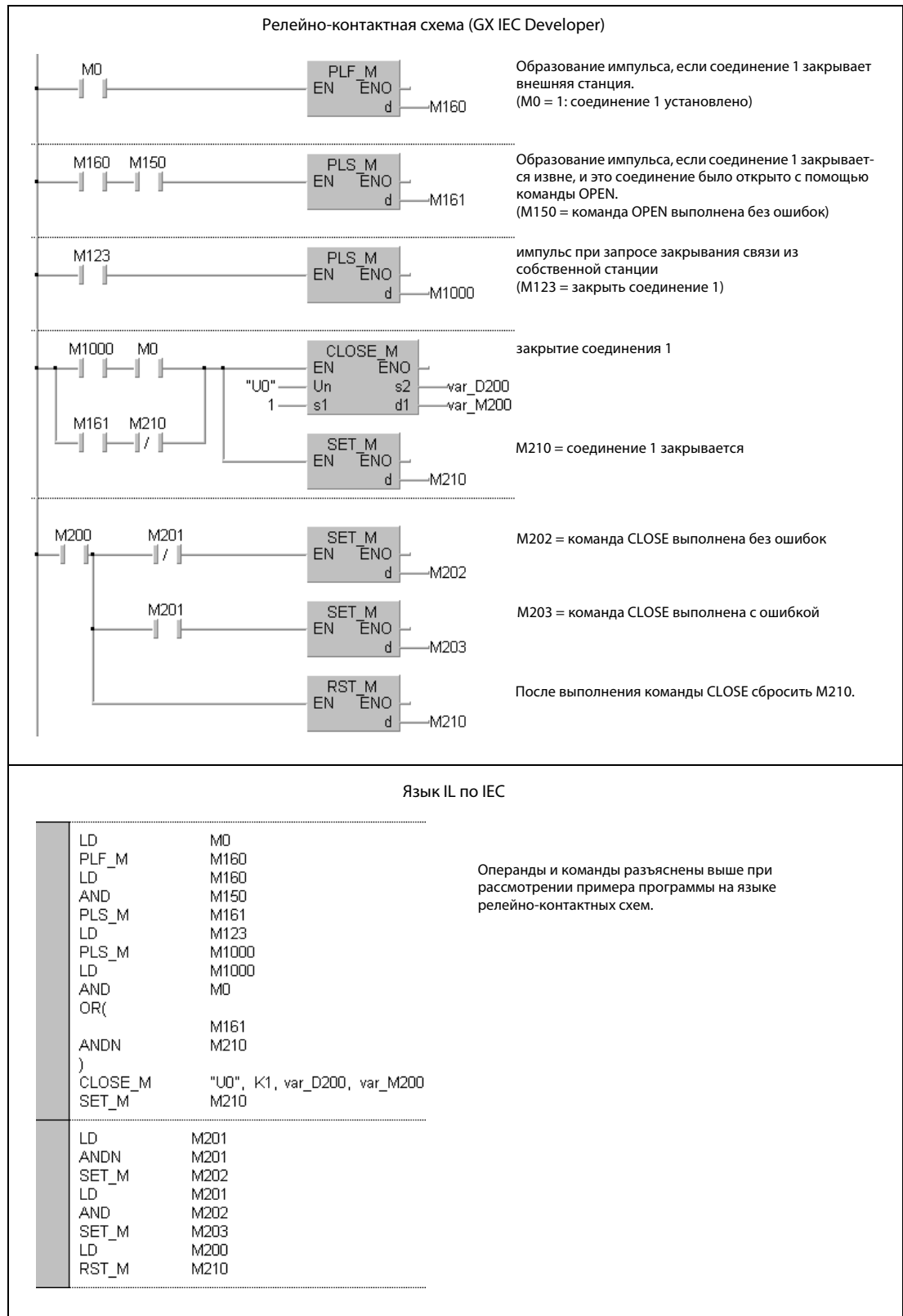
- При коде ошибки до  $4FFF_H$  указания по устранению ошибки имеются в этом руководстве (глава 13).
- При коде ошибки, начиная с  $C001_H$ , более подробная информация содержится в руководствах по модулям Ethernet для серии "System Q".

**Пример**

CLOSE

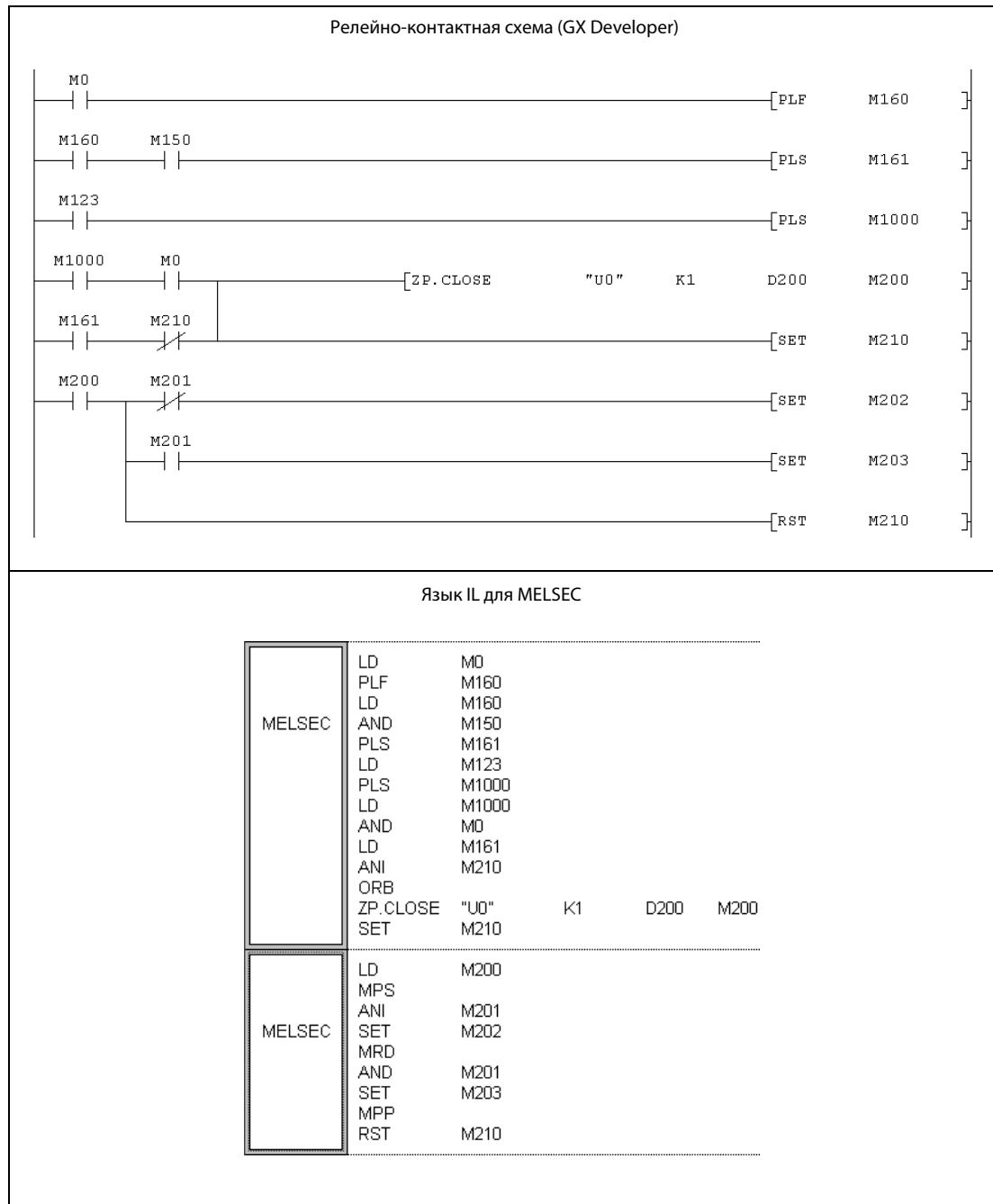
Следующая программа закрывает соединение 1 модуля Ethernet, имеющего головной адрес X/Y0.

- Редакторы по стандарту IEC (на следующей странице этот пример воспроизведен на языке списка инструкций MELSEC и языке LD в среде GX Developer)



**ПРИМЕЧАНИЕ** В редакторах по стандарту IEC среды GX IEC Developer переменные должны быть объявлены в заголовке программного компонента (POU). Если переменные не были объявлены, то при проверке или компилировании программы возникает сообщение об ошибке. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

- Список инструкций MELSEC и релейно-контактная схема в среде GX Developer  
Операнды и команды разъяснены на предыдущей странице при рассмотрении примера программы на языке LD в среде GX IEC Developer.



### 11.3.6 ERRCLR

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

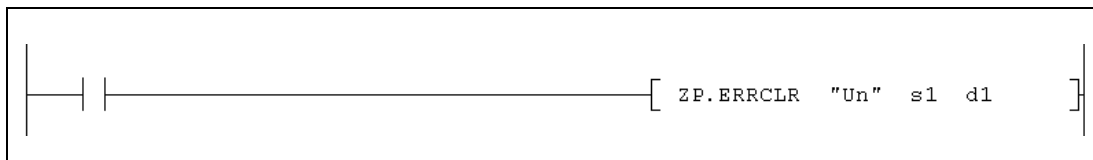
Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	—	SM0	
d1	●	●	●	—	—	—	—	—	—		

GX IEC  
Developer



GX  
Developer



## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных	
„Un“	Головной адрес модуля Ethernet на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде "U10".)	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит	
s1	Первый операнд области, содержащей информацию для выполнения команды.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(s1) + 0	Системная область	Используется системой		
	(s1) + 1	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды: 0000 <sub>H</sub> : безошибочная обработка Любое иное значение кроме 0000 <sub>H</sub> : При обработке возникла ошибка. Записанное значение представляет собой код ошибки, который разъяснен либо в руководстве по модулю Ethernet, либо в глава 13 этого руководства.	—	система
	(s1) + 2	Стираемая ошибка	В зависимости от введенного значения стирается код ошибки, сохраненный в буферной памяти модуля, и/или выключается светодиод "ERR." модуля Ethernet. Подробное описание имеется ниже.	0000 <sub>H</sub> 0001 <sub>H</sub> до 0016 <sub>H</sub> 0100 <sub>H</sub> 0101 <sub>H</sub> 0102 <sub>H</sub> 0103 <sub>H</sub> FFFF <sub>H</sub>	пользователь
	(s1) + 3	Функция	Выбор между выключением светодиода и стиранием кода ошибки. Функции более подробно описаны ниже.	0000 <sub>H</sub> или FFFF <sub>H</sub>	
от (s1) + 4 до (s1) + 7	Системная область	Используется системой.	—	система	
d1	Битовый операнд, устанавливаемый на один цикл после выполнения команды ERRCLR. По адресу (d1) + 1 выводится ошибка при выполнении.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(d1) + 0	Команда выполнена	Показывает завершение команды ERRCLR. Вкл.: команда выполнена Выкл.: команда не выполнена	—	
(d1) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды ERRCLR. Вкл.: команда выполнена с ошибкой Выкл.: команда выполнена без ошибок	—	система	

**Принцип действия****Стирание ошибки в буферной памяти и выключение светодиода "ERR."****ERRCLR Стирание ошибки**

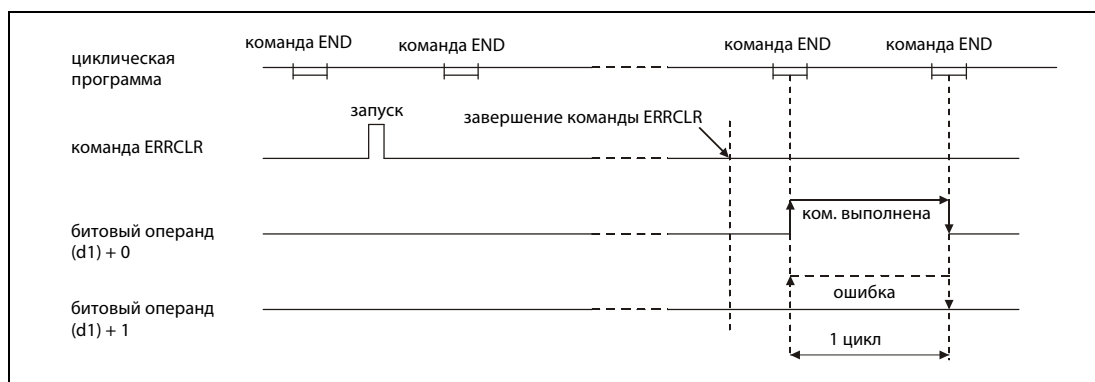
Команда ERRCLR стирает код ошибки, сохраненный в буферной памяти модуля Ethernet. Если светодиод "ERR." с передней стороны модуля горит, то после выполнения команды ERRCLR он выключается. Кроме того, с помощью этой команды можно стереть области буферной памяти, в которых указано состояние коммуникации.

Стираемую область в буферной памяти определяет содержимое операндов (s1) + 2 и (s1) + 3:

Ошибка или область состояния		Содержимое операндов		Выполненное действие
		(s1) + 2	(s2) + 3	
Ошибка при запуске модуля		0000 <sub>H</sub>	0000 <sub>H</sub>	<ul style="list-style-type: none"> <li>• Стирается адрес 69<sub>H</sub> буферной памяти.</li> <li>• Выключается светодиод "ERR."</li> </ul>
Ошибка при открытии связи		0001 <sub>H</sub> ... 0016 <sub>H</sub> (номер соедин.)		<ul style="list-style-type: none"> <li>• Стирание адреса в буферной памяти, по которому был записан код ошибки соединения (7C<sub>H</sub>, 86<sub>H</sub> ...).</li> <li>• Выключается светодиод "ERR."</li> </ul>
Буфер сообщений об ошибках		0100 <sub>H</sub>	FFFF <sub>H</sub>	<ul style="list-style-type: none"> <li>• Стирается буфер сообщений об ошибках (адреса с E3<sub>H</sub> по 174<sub>H</sub>).</li> </ul>
Состояние коммуникации	Состояние протоколов	0101 <sub>H</sub>		<ul style="list-style-type: none"> <li>• Стирается область буферной памяти от 178<sub>H</sub> до 1FF<sub>H</sub>.</li> </ul>
	Состояние приема электронной почты	0102 <sub>H</sub>		<ul style="list-style-type: none"> <li>• Стирание области буферной памяти от 5B71<sub>H</sub> до 5B38<sub>H</sub>.</li> </ul>
	Состояние отправки электронной почты	0103 <sub>H</sub>		<ul style="list-style-type: none"> <li>• Стирание области буферной памяти от 5B39<sub>H</sub> до 5CA0<sub>H</sub>.</li> </ul>
Все сохраненные ошибки или области состояния		FFFF <sub>H</sub>	<ul style="list-style-type: none"> <li>• Стираются все вышеупомянутые адреса буферной памяти.</li> <li>• Выключается светодиод "ERR."</li> </ul>	

Выполнение команды ERRCLR можно проверить по состоянию битовых операндов (d1) + 0 и (d1) + 1:

- Битовый операнд (d1) + 0 устанавливается при выполнении команды END того цикла, в котором команда ERRCLR была завершена. При обработке следующей команды END указанный в (d1) + 0 битовый операнд снова сбрасывается.
- Битовый операнд (d1) + 1 показывает ошибку при выполнении команды ERRCLR на. При безошибочном выполнении этот битовый операнд остается сброшенным. В случае ошибки (d1) + 1 устанавливается при выполнении команды END того цикла, в котором команда ERRCLR была завершена. При обработке следующей команды END этот операнд снова сбрасывается.



**Источники  
ошибок**

Если при выполнении команды ERRCLR возникла ошибка, то устанавливается операнд (d1) + 1 и в операнд (s1) + 1 записывается код ошибки. Коды ошибок подробно разъяснены в следующих руководствах:

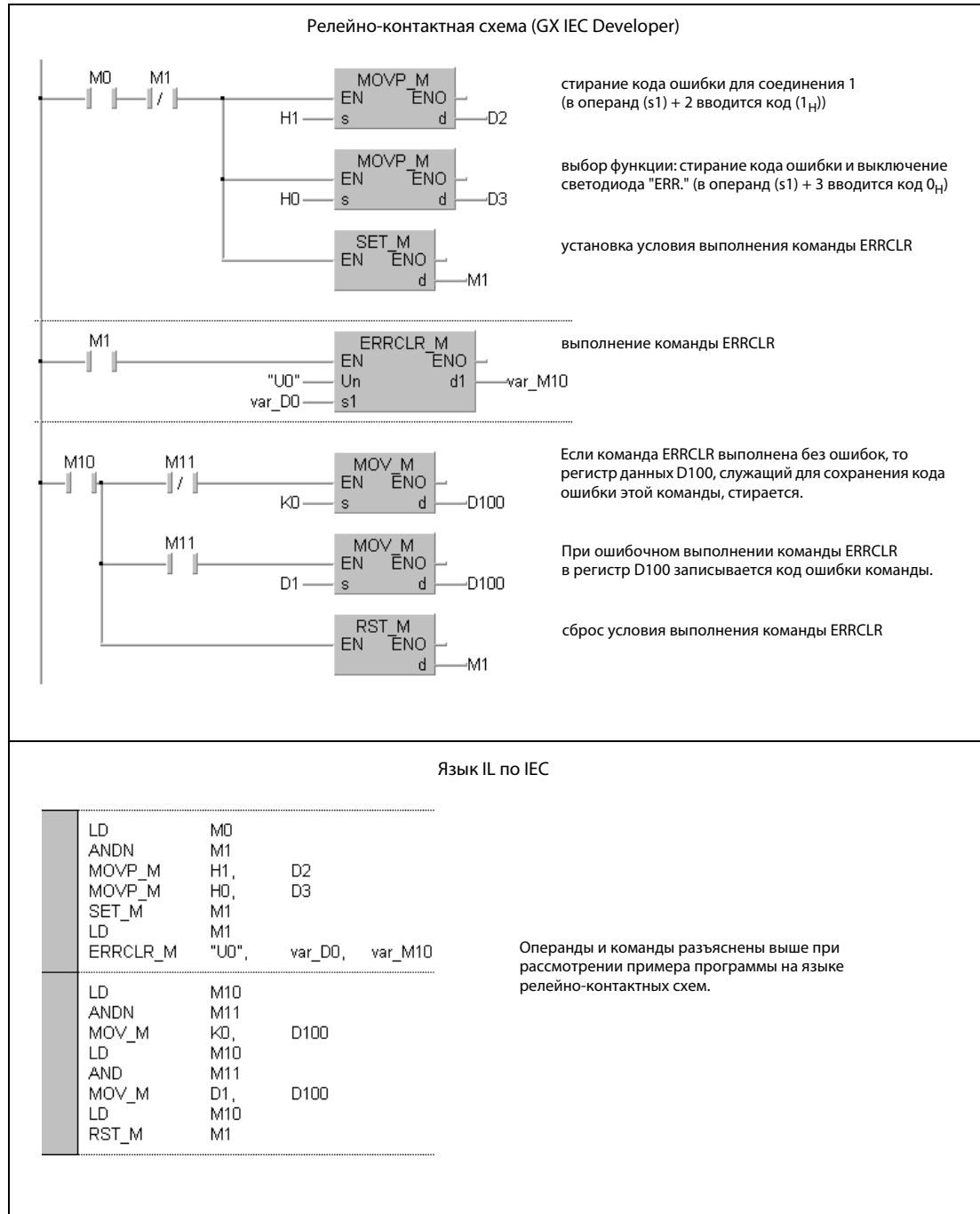
- При коде ошибки до 4FFF<sub>H</sub> указания по устранению ошибки имеются в этом руководстве (глава 13).
- При коде ошибки начиная с C001<sub>H</sub> более подробная информация содержится в руководствах по модулям Ethernet для серии "System Q".



**Пример** ERRCLR

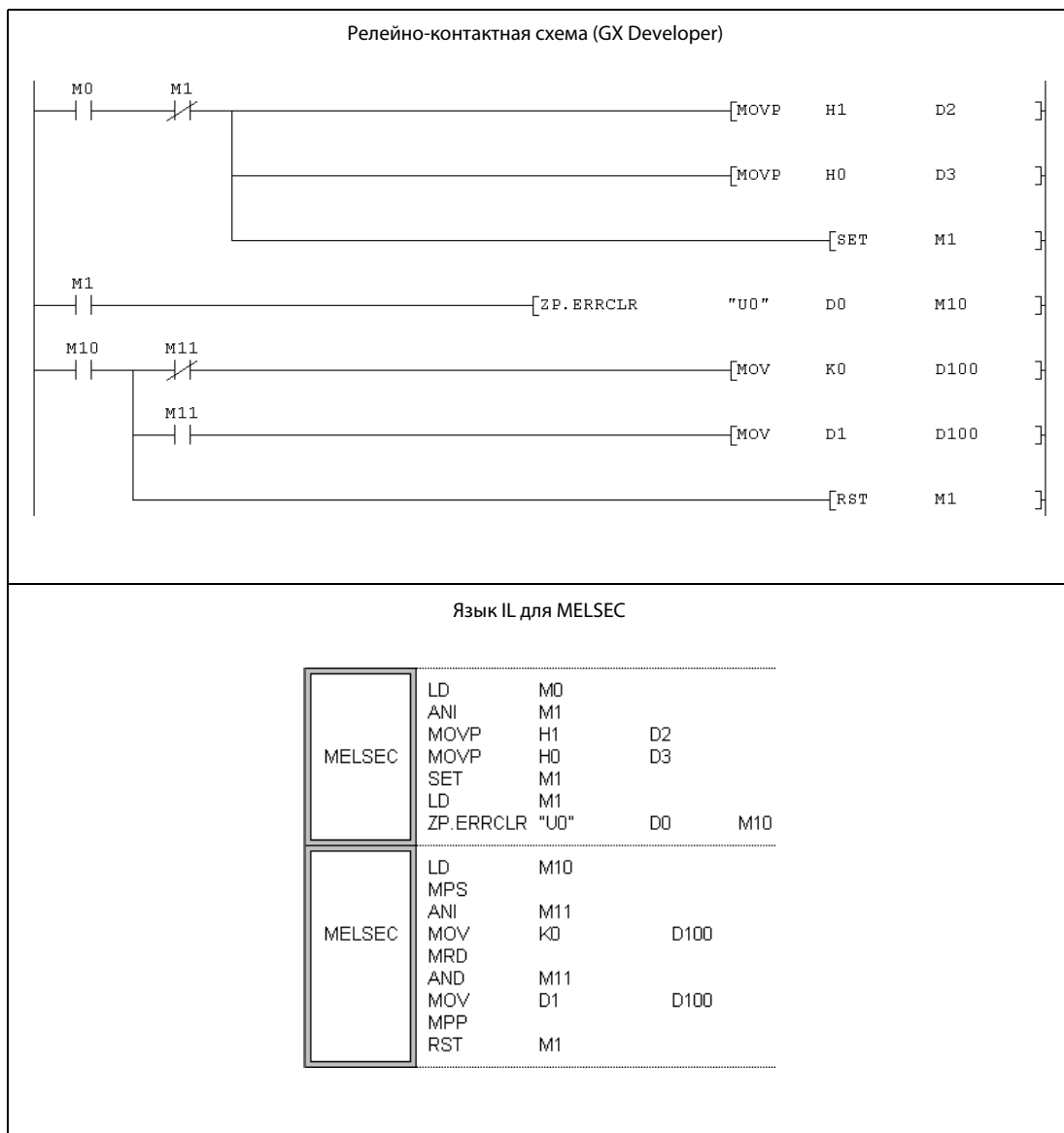
Следующая программа стирает код ошибки, зарегистрированный для соединения 1. Модуль Ethernet имеет головной адрес X/Y0.

- Редакторы по стандарту IEC (на следующей странице этот пример воспроизведен на языке списка инструкций MELSEC и языке LD в среде GX Developer)



**ПРИМЕЧАНИЕ** В редакторах по стандарту IEC среды GX IEC Developer переменные должны быть объявлены в заголовке программного компонента (POU). Если переменные не были объявлены, то при проверке или компиляции программы возникает сообщение об ошибке. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

- Список инструкций MELSEC и релейно-контактная схема в среде GX Developer  
 Операнды и команды разъяснены на предыдущей странице при рассмотрении примера программы на языке LD в среде GX IEC Developer.



### 11.3.7 ERRRD

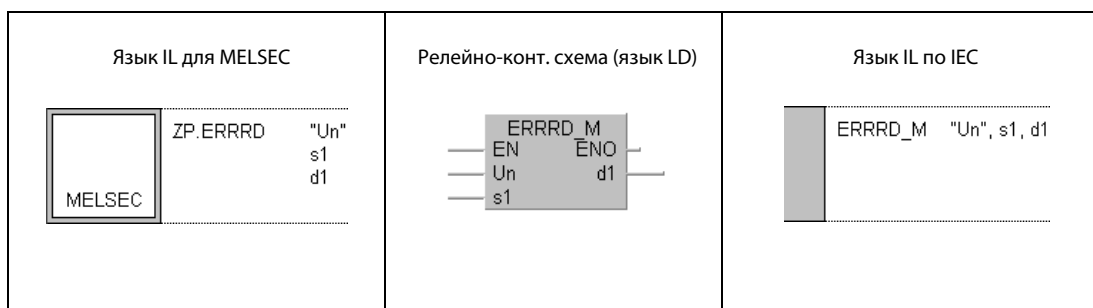
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

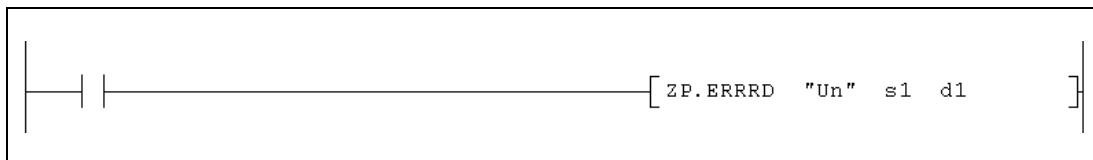
Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	—	SM0	
d1	●	●	●	—	—	—	—	—	—		

GX IEC  
Developer



GX  
Developer



## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных	
„Un“	Головной адрес модуля Ethernet на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде "U10".)	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит	
s1	Первый операнд области, содержащей информацию для выполнения команды.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(s1) + 0	Системная область	Используется системой		
	(s1) + 1	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды: 0000 <sub>H</sub> : безошибочная обработка Любое иное значение кроме 0000 <sub>H</sub> : При обработке возникла ошибка. Записанное значение представляет собой код ошибки, который разъяснен либо в руководстве по модулю Ethernet, либо в глава 13 этого руководства.	—	система
	(s1) + 2	Считываемый код ошибки	В зависимости от введенного значения считывается следующий код ошибки, сохраненный в буферной памяти модуля Ethernet: • 0000 <sub>H</sub> : код, который был записан при ошибке запуска модуля в адрес 69 <sub>H</sub> буферной памяти • 0001 <sub>H</sub> ... 0016 <sub>H</sub> : код ошибки для соответствующего соединения (адрес в буферной памяти 7C <sub>H</sub> , 86 <sub>H</sub> ...)	0000 <sub>H</sub> 0001 <sub>H</sub> ... 0016 <sub>H</sub>	пользователь
	(s1) + 3	Функция	Считывание кодов ошибок, возникших последними	0000 <sub>H</sub>	
	(s1) + 4	Считанный код ошибки	Содержит код ошибки, переданный модулем Ethernet. 0000 <sub>H</sub> = ошибок нет Иные значения кроме 0000 <sub>H</sub> : код ошибки	—	система
	от (s1) + 5 до (s1) + 7	Системная область	Используется системой.	—	система
d1	Битовый операнд, устанавливаемый на один цикл после выполнения команды ERRRD. По адресу (d1) + 1 выводится ошибка при выполнении.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(d1) + 0	Команда выполнена	Показывает завершение команды ERRRD. ВКЛ.: команда выполнена ВЫКЛ.: команда не выполнена	—	
(d1) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды ERRRD. ВКЛ.: команда выполнена с ошибкой ВЫКЛ.: команда выполнена без ошибок	—	система	

**Принцип действия****Считывание кода ошибки из модуля Ethernet****ERRRD Считывание кода ошибки**

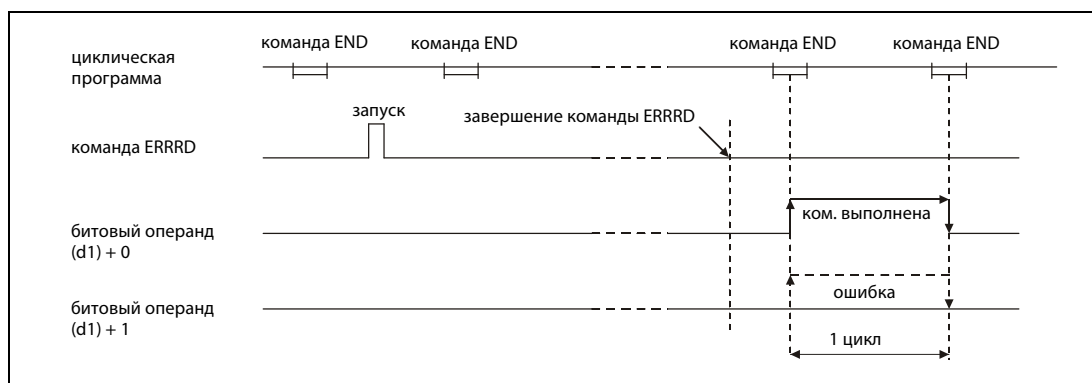
Команда ERRCLR считывает код ошибки, сохраненный в буферной памяти модуля Ethernet.

В операнде (s1) + 2 указывается, из какого адреса буферной памяти требуется считать код ошибки. В "Un" указывается головной адрес модуля Ethernet.

Выполнение команды ERRRD можно проверить по состоянию битовых операндов (d1) + 0 и (d1) + 1:

- Битовый операнд (d1)+0 устанавливается при выполнении команды END того цикла, в котором команда ERRRD была завершена. При обработке следующей команды END этот битовый операнд снова сбрасывается.
- Битовый операнд (d1) + 1 показывает ошибку при выполнении команды ERRRD. При безошибочном выполнении этот битовый операнд остается сброшенным. В случае ошибки (d1) + 1 устанавливается при выполнении команды END того цикла, в котором команда ERRRD была завершена. При обработке следующей команды END этот операнд снова сбрасывается.

На рисунке ниже показана диаграмма изменения сигналов при выполнении команды ERRRD:

**Источники ошибок**

Если при выполнении команды ERRRD возникла ошибка, то устанавливается операнд (d1) + 1 и код записывается ошибки в операнд, указанный в (s1) + 1. Коды ошибок подробно разъяснены в следующих руководствах:

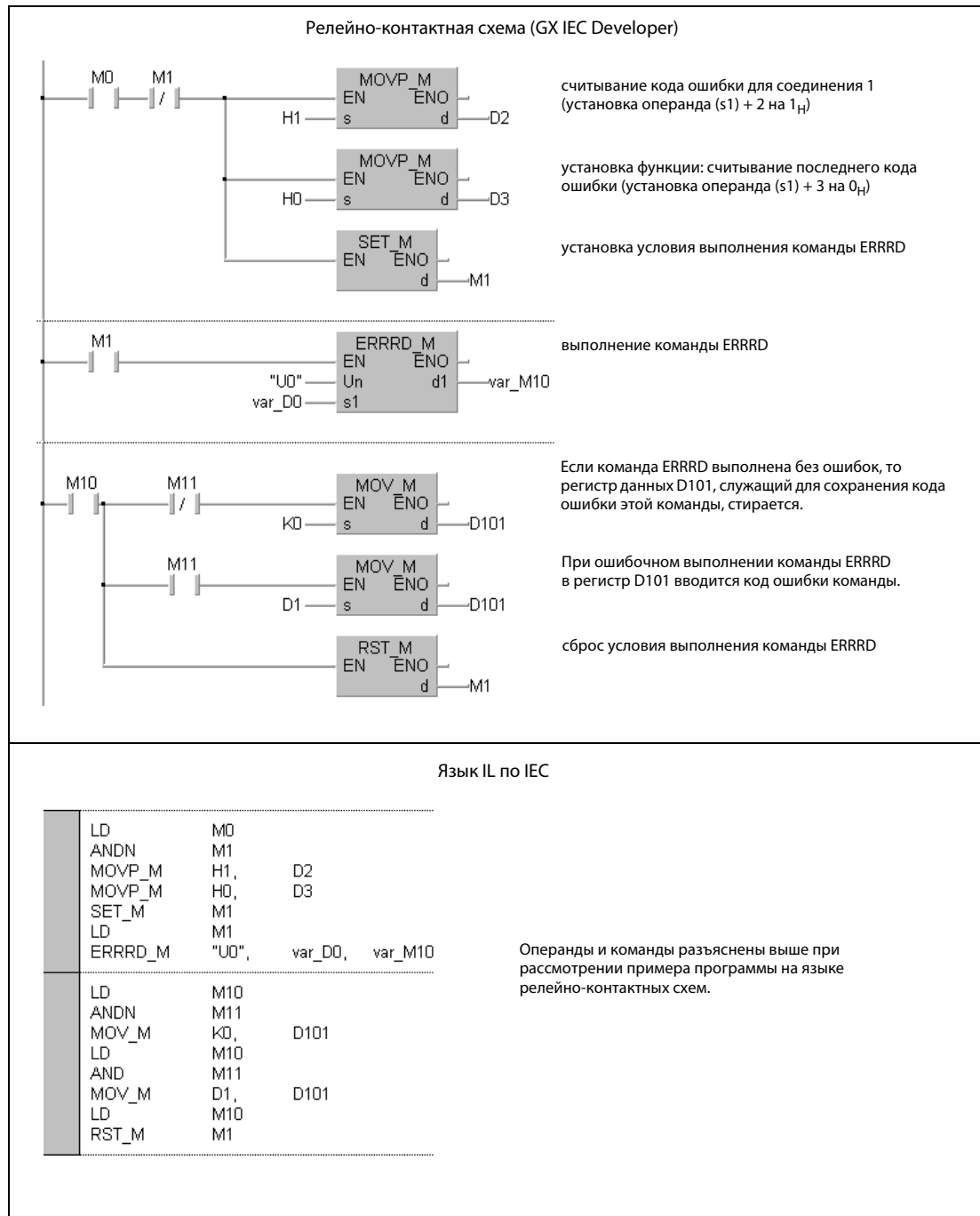
- При коде ошибки до 4FFF<sub>H</sub> указания по устранению ошибки имеются в этом руководстве (глава 13).
- При коде ошибки, начиная с C001<sub>H</sub>, более подробная информация содержится в руководствах по модулям Ethernet для серии "System Q".

## Пример

## ERRRD

Следующая программа считывает код ошибки, если при открывании соединения 1 возникла ошибка. Модуль Ethernet имеет головной адрес X/Y0.

- Редакторы по стандарту IEC (на следующей странице этот пример воспроизведен на языке списка инструкций MELSEC и языке LD в среде GX Developer)

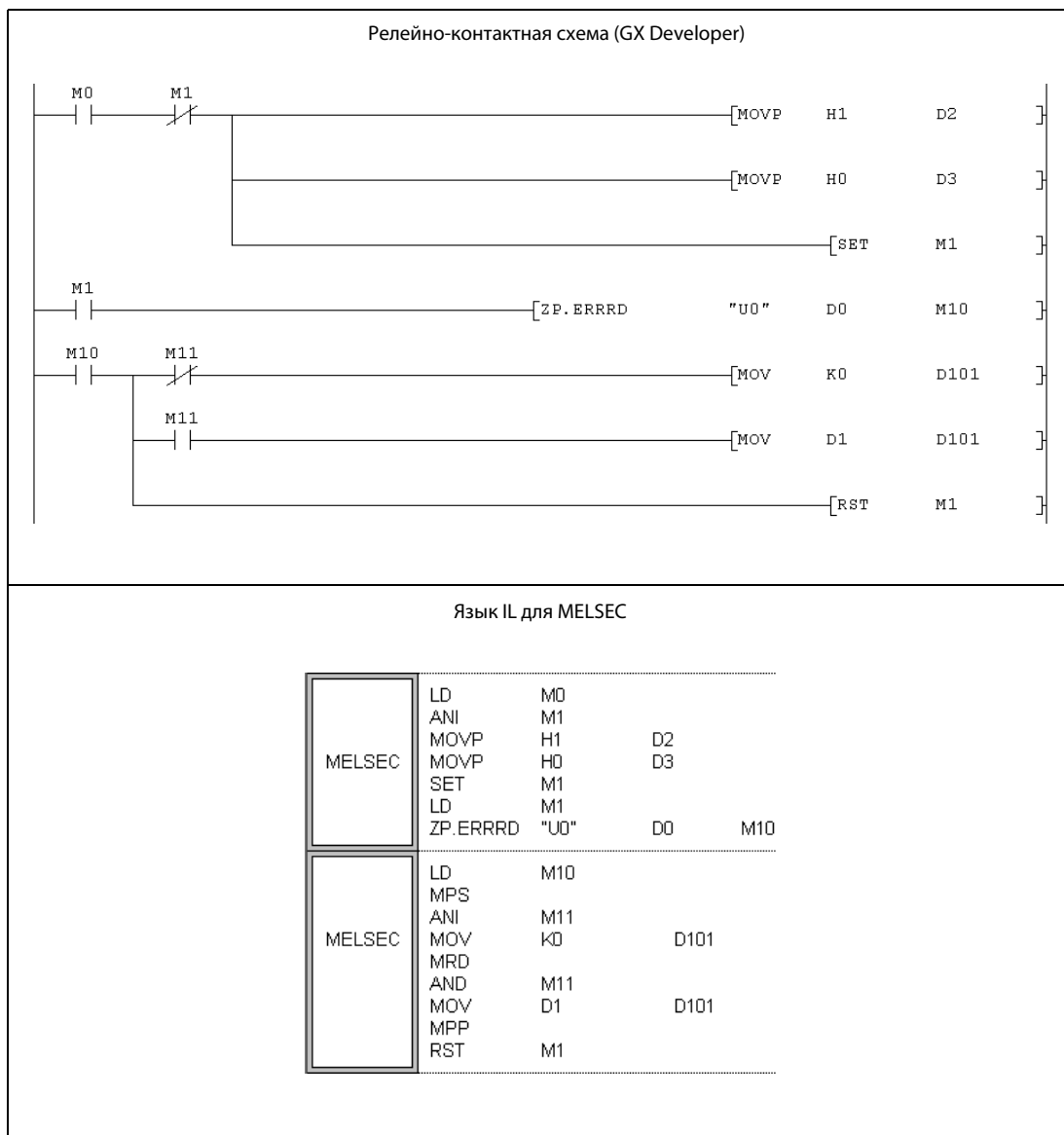


## ПРИМЕЧАНИЕ

В редакторах по стандарту IEC среды GX IEC Developer переменные должны быть объявлены в заголовке программного компонента (POU). Если переменные не были объявлены, то при проверке или компиляции программы возникает сообщение об ошибке.

Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

- Список инструкций MELSEC и релейно-контактная схема среды GX Developer  
Операнды и команды разъяснены на предыдущей странице при рассмотрении примера программы на языке LD в среде GX IEC Developer.



### 11.3.8 UINI

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

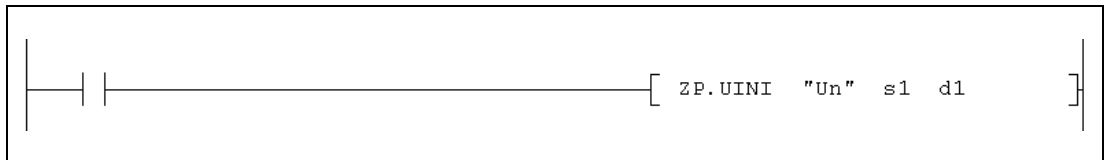
Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□V□G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	SM0		
d1	●	●	●	—	—	—	—	—			

GX IEC  
Developer



GX  
Developer



Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных
„Un“	Головной адрес модуля Ethernet на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде "U10")	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит



## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных		
s1	Первый операнд области, содержащей информацию для выполнения команды				BIN, 16 бит	
	Операнд	Значение	Описание	Диапазон		Устанавливает
	(s1) + 0	Системная область	Используется системой			
	(s1) + 1	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды: 0000 <sub>H</sub> : безошибочная обработка Любое иное значение кроме 0000 <sub>H</sub> : При обработке возникла ошибка. Записанное значение представляет собой код ошибки, который разъяснен либо в руководстве по модулю Ethernet, либо в глава 13 этого руководства.	—		система
	(s1) + 2	Выбор изменений	Биты 0 и 1 этого словного операнда служат для настройки свойств соединения. <ul style="list-style-type: none"> <li>Бит 0: изменение IP-адреса локальной станции (новый адрес указывается в (s1) + 3 и (s1) + 4.) 0: не изменять IP-адрес 1: изменить IP-адрес</li> <li>Бит 1: изменение рабочих настроек (новые настройки указываются в (s1) + 5.) 0: не изменять настройки 1: изменить настройки</li> </ul> Прочие биты (от b2 до b15) следует установить на "0".	0000 <sub>H</sub> ... 0003 <sub>H</sub>		
	(s1) + 3 (s1) + 4	IP-адрес локальной станции	Новый IP-адрес локальной станции	00000001 <sub>H</sub> ... FFFFFFFE <sub>H</sub>		
(s1) + 5	Рабочие настройки	С помощью отдельных битов этого словного операнда устанавливаются настройки. <ul style="list-style-type: none"> <li>Бит 1: кодировка данных при коммуникации 0: двоичная кодировка 1: ASCII-кодировка</li> <li>Бит 5: кадр передачи 0: кадр Ethernet 1: кадр по IEEE802.3</li> <li>Бит 6: деблокировка изменения программы центрального процессора в режиме RUN 0: изменения не допускаются 1: изменения допускаются</li> <li>Бит 8: время ожидания 0: не ждать открытия соединения (при остановленном центральном процессоре коммуникация не возможна) 1: всегда ждать открытия соединения (коммуникация при остановленном центральном процессоре возможна)</li> </ul> Прочие биты этого операнда должны быть установлены на "0".	—	пользователь		

**Переменные**

Операнд	Значение			Диапазон	Устанавливает	Тип данных
d1	Битовый операнд, устанавливаемый на один цикл после выполнения команды UINI. По адресу (d1) + 1 выводится ошибка при выполнении.					бит
	Операнд	Значение	Описание	Диапазон	Устанавливает	
	(d1) + 0	Команда выполнена	Показывает завершение команды UINI. Вкл.: команда выполнена Выкл.: команда не выполнена	—	система	
(d1) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды UINI. Вкл.: команда выполнена с ошибкой Выкл.: команда выполнена без ошибок	—			

**ПРИМЕЧАНИЕ**

Если модуль Ethernet требуется лишь заново инициализировать (без изменения IP-адреса локальной станции или рабочих настроек), то перед выполнением команды UINI в операнд (s1) + 2 необходимо записать значение "0".  
При инициализации в модуле Ethernet стирается информация об адресах других станций и снова становится возможным обмен данными. По окончании инициализации включается вход X19.

**Принцип действия**

**Новая инициализация модуля Ethernet**

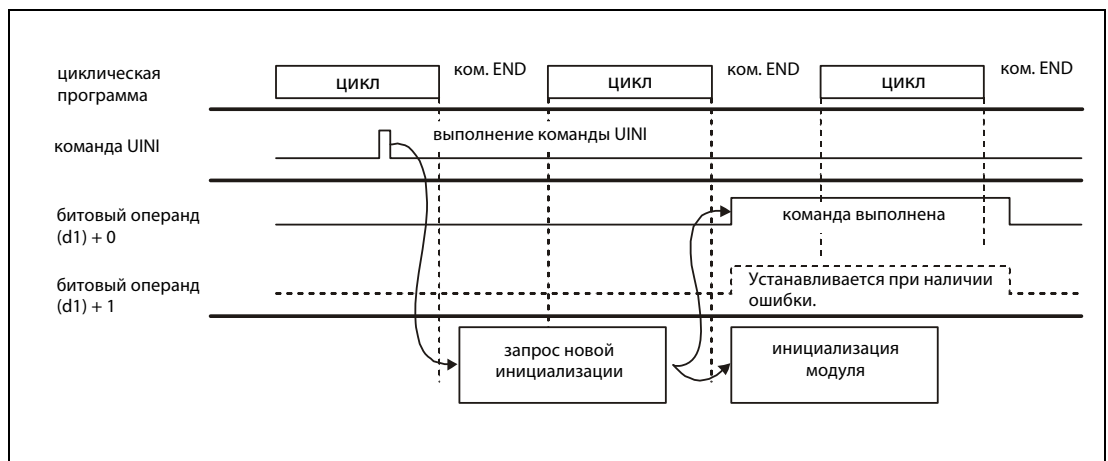
**UINI      Запуск инициализации**

Команда UINI заново инициализирует модуль Ethernet, указанный в Un.

Выполнение команды UINI можно проверить по состоянию битовых операндов (d1) + 0 и (d1) + 1:

- Битовый операнд (d1) + 0 устанавливается при выполнении команды END того цикла, в котором команда UINI была завершена. При обработке следующей команды END указанный в d1 битовый операнд снова сбрасывается.
- Битовый операнд (d1) + 1 показывает ошибку при выполнении команды UINI. При безошибочном выполнении этот битовый операнд остается сброшенным. В случае ошибки (d1) + 1 устанавливается при выполнении команды END того цикла, в котором команда UINI была завершена. При обработке следующей команды END этот операнд снова сбрасывается.

На рисунке ниже показана диаграмма изменения сигналов при выполнении команды UINI:



**ПРИМЕЧАНИЯ** При повторной инициализации модуля Ethernet учитывайте следующие пункты. (В противном случае могут возникнуть ошибки при обмене данными с внешними модулями.)

– Прежде чем заново инициализировать модуль Ethernet, убедитесь в том, что все происходившие процессы коммуникации с внешними модулями завершены, и закройте все соединения.

– Во время выполнения команды UINI не разрешается записывать данные непосредственно в буферную память (например, с помощью команды TO).

Не запрашивайте еще одну новую инициализацию, в то время как уже выполняется команда UINI.

– Если изменяется IP-адрес модуля Ethernet, должен быть выполнен сброс внешних модулей. (Если внешний модуль сохранит адрес Ethernet модуля, с которым он коммуницирует, то после изменения IP-адреса коммуникация не сможет быть продолжена.)

**Источники ошибок**

Если при выполнении команды UINI возникла ошибка, то устанавливается операнд (d1) + 1 и код ошибки записывается в операнд (s1) + 1. Подробная информация о кодах ошибок имеется в следующих руководствах:

- При коде ошибки до 4FFF<sub>H</sub> указания по устранению ошибки имеются в этом руководстве (глава 13).
- При коде ошибки, начиная с C001<sub>H</sub>, более подробная информация содержится в руководствах по модулям Ethernet для серии "System Q".

**Пример**

UINI

Следующая программа заново инициализирует модуль Ethernet с головным адресом X/Y0 (диапазон адресов от X/Y0 до X/Y1F).

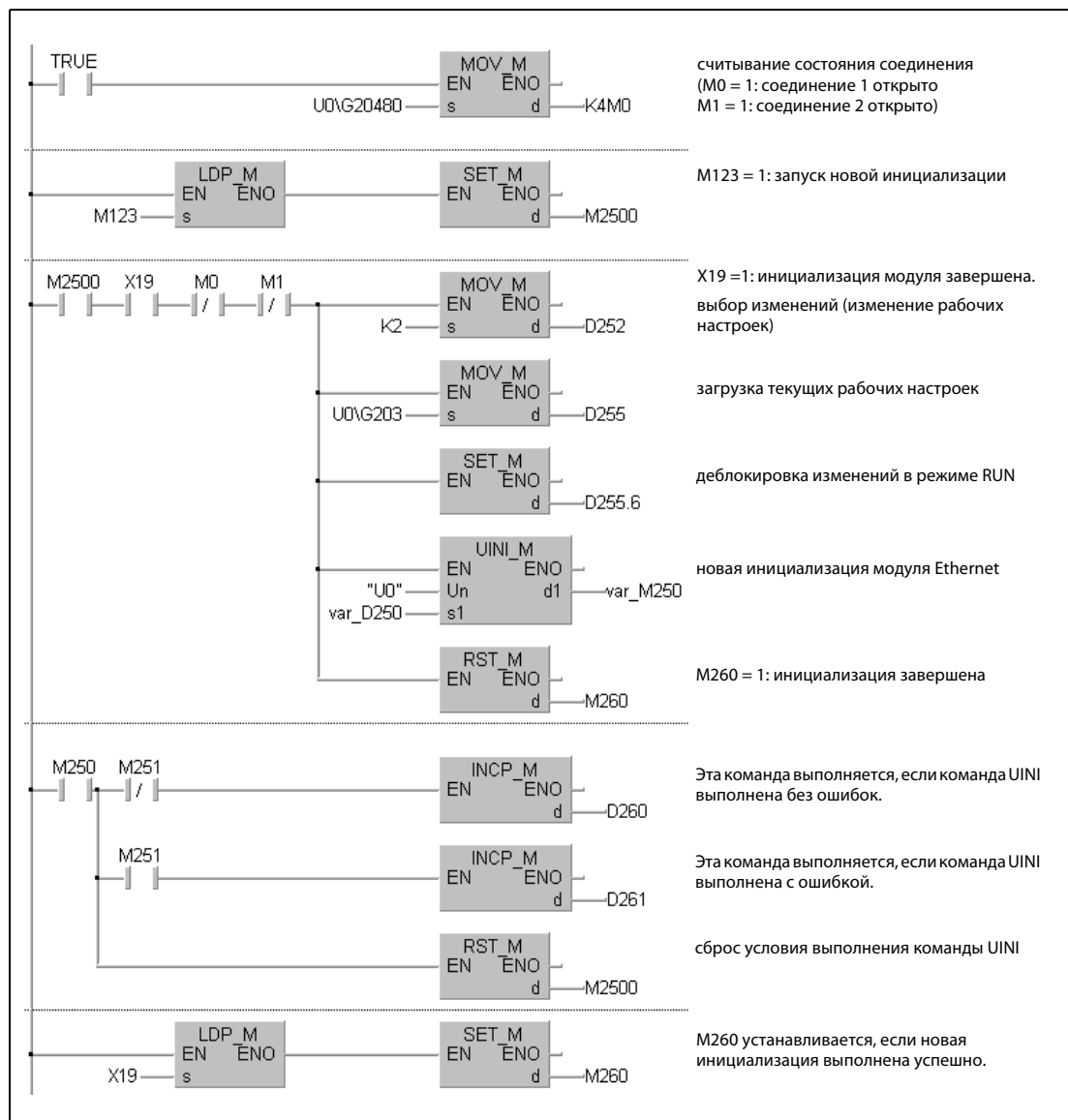
**ПРИМЕЧАНИЯ**

В этом примере программы используются только соединения 1 и 2. В случае других соединений в программе должны обрабатываться соответствующие сигналы.

В редакторах по стандарту IEC среды GX IEC Developer переменные должны быть объявлены в заголовке программного компонента (POU). Если переменные не были объявлены, то при проверке или компилировании программы возникает сообщение об ошибке.

Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

● Релейно-контактная схема (GX IEC Developer)



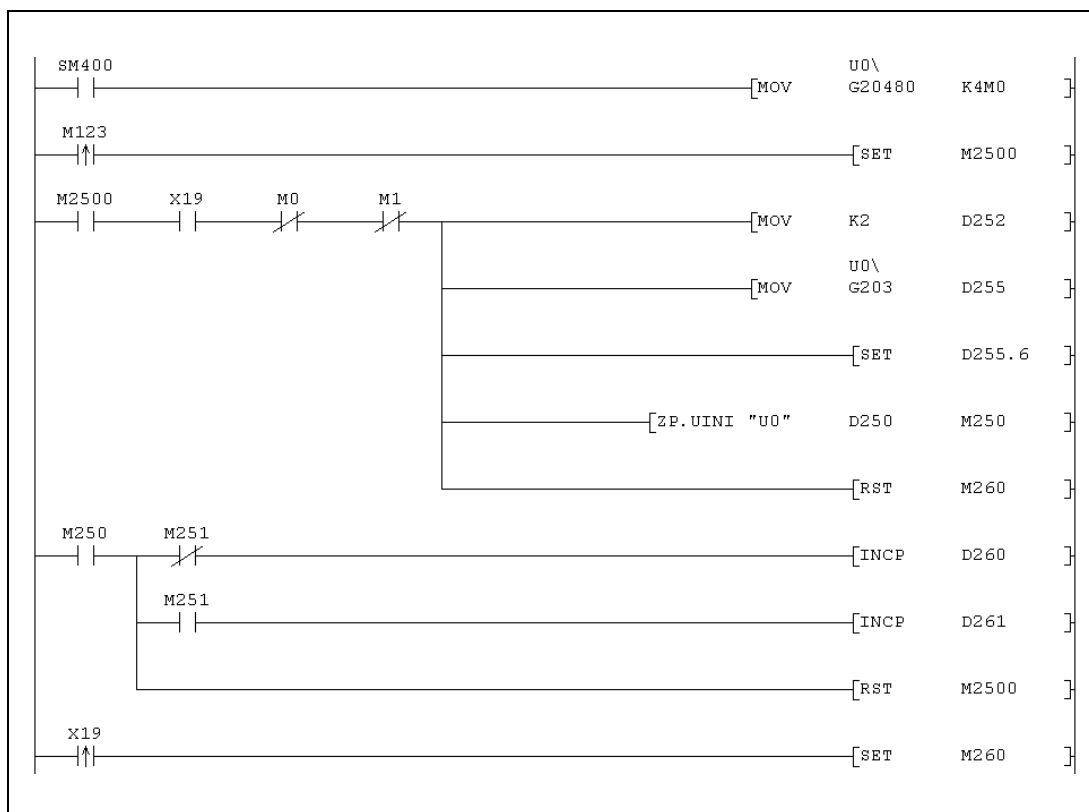
● Язык IL по стандарту IEC и язык IL для MELSEC

Операнды и команды разъяснены при рассмотрении примера программы на языке LD в среде GX IEC Developer.

Язык IL по IEC			Язык IL для MELSEC			
LD	TRUE		MELSEC	LD	SM400	
MOV_M	U0\G20480, K4M0			MOV	U0\G20480	K4M0
PLS_M	M123			LDP	M123	
SET_M	M2500			SET	M2500	
AND	M2500			LD	M2500	
AND	X19			AND	X19	
ANDN	M0			ANI	M0	
ANDN	M1			ANI	M1	
MOV_M	K2, D252			MOV	K2	D252
MOV_M	U0\G203, D255			MOV	U0\G203	D255
SET_M	D255.6			SET	D255.6	
UINI_M	"U0", var_D250, var_M250			ZP.UINI	"U0"	D250 M250
RST_M	M260			RST	M260	
LD	M250		MELSEC	LD	M250	
ANDN	M251			MPS		
INCP_M	D260			ANI	M251	
LD	M250			INCP	D260	
AND	M251			MRD		
INCP_M	D261			AND	M251	
LD	M250			INCP	D261	
RST_M	M2500			MPP		
PLS_M	X19			RST	M2500	
SET_M	M260			LDP	X19	
				SET	M260	

● Релейно-контактная схема (GX Developer)

Операнды и команды разъяснены при рассмотрении примера программы на языке LD в среде GX IEC Developer.



## 11.4 Команды для MELSECNET/10

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту IEC
Указание станций для дуплексного режима	J.PAIRSET	PAIRSET_M

### 11.4.1 PAIRSET

**Процессор**

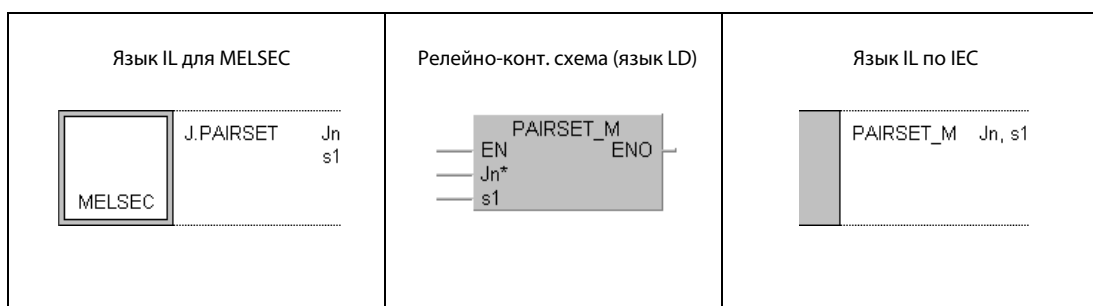
AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				● <sup>1</sup>	

<sup>1</sup> только для Q4AR

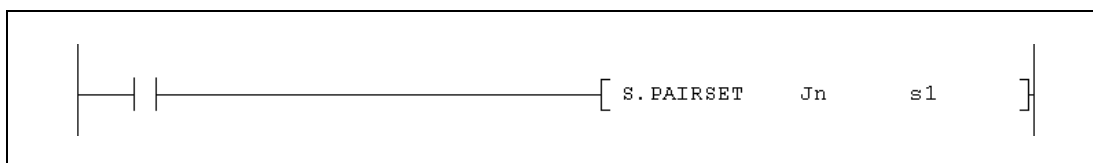
**Операнды MELSEC Q**

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непос. адр. J□□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	—	S0	

**GX IEC Developer**



**GX Developer**



Операнд	Значение	Тип данных
Jn	Номер сети (от 1 до 239)	BIN, 16 бит
s1	Начальный адрес области операндов, в которой сохраняются настройки для пар. Можно использовать регистр файлов (R, ZR) или операнды T, ST, C, D и W из области фиксации. В случае использования регистров файлов необходима карта памяти.	

**Принцип действия**

**Установка пар станций**

**PAIRSET Установка пар**

Эта команда указывает область операндов, в которой установлено, какие станции соединены при дуплексном режиме.

**Структура области операндов, содержащей настройки**

- Установить станции в операнде, указанном в s1, с помощью основной программы не возможно. Эти данные должны быть заранее сохранены в центральном процессоре контроллера с помощью программатора.
- Независимо от количества подключенных станций используются четыре слова.
- Образовать пару можно только из станций со следующими друг за другом номерами. Для образования пары должен быть установлен бит, который в области операндов, обозначенной в s1, указывает станцию с более высоким номером.
- Каждый бит в области операндов от (s1) + 0 до (s1) + 3 означает номер станции в диапазоне между 1 и 64:

Операнд	Биты															
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
(s1) + 0	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
(s1) + 1	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
(s1) + 2	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
(s1) + 3	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49

**ПРИМЕЧАНИЯ**

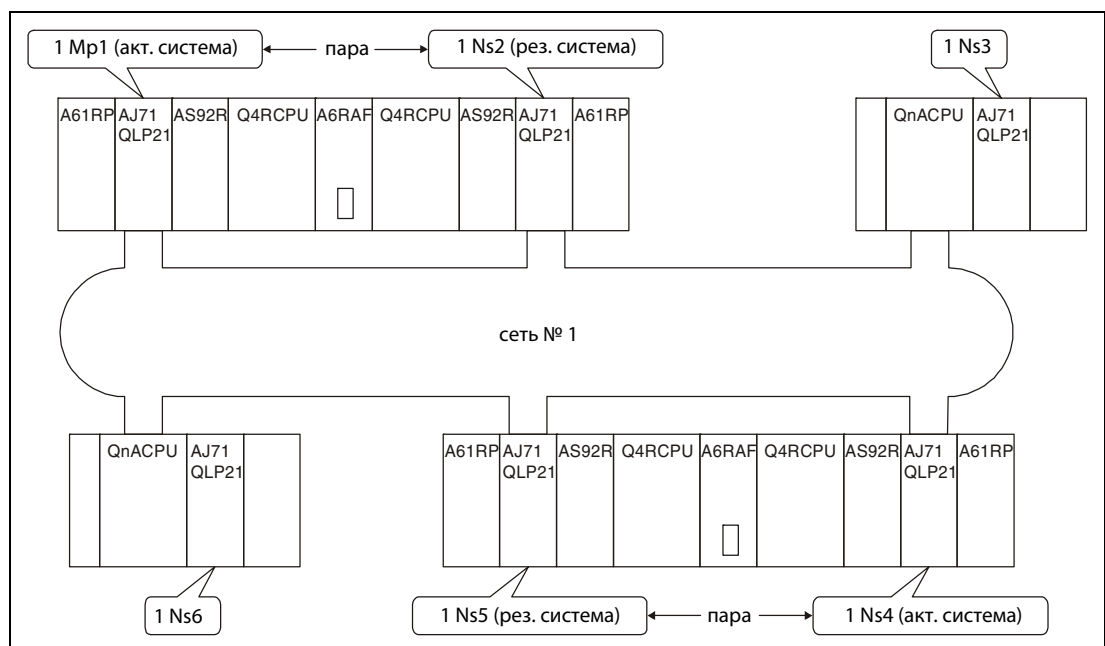
Команда PAIRSET действует только на управляющих станциях. Настройки на обычной станции недействительны.

В избыточной системе управления на центральных процессорах Q4AR, если коммуникация с сетевым модулем активной системы стала невозможной из-за обрыва провода, переключение на резервную систему происходит только в том случае, если была выполнена команда PAIRSET.

**Пример**

**PAIRSET**

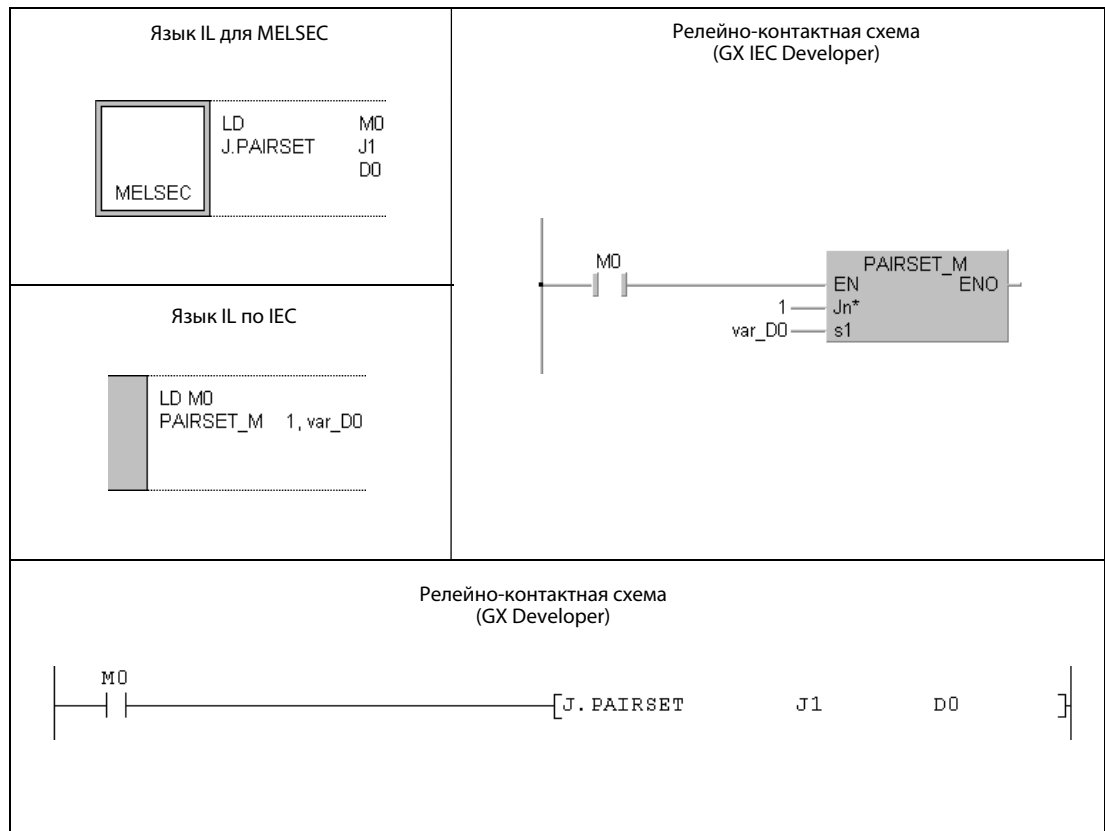
Требуется образовать пары из станций 1 и 2, а также станций 4 и 5 избыточной системы:





Настройки для пар заложены в регистрах данных с D0 по D3. В D0 устанавливается бит b1 (для соединения станций 1 и 2) и b4 (для соединения станций 4 и 5):

Операнд	Биты															
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
D0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



**ПРИМЕЧАНИЕ** В редакторах по стандарту IEC среды GX IEC Developer переменные должны быть объявлены в заголовке программного компонента (POU). Если переменные не были объявлены, то при проверке или компиляции программы возникает сообщение об ошибке. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

## 11.5 Команды для CC-Link

Функция	Команда MELSEC в редакторе MELSEC	Команда MELSEC в редакторе по стандарту IEC
Передача параметров сети в мастер-станцию (серия "A")	G.RLPA	RLPA_MD
	GP.RLPA	RLPA_P_MD
Передача параметров сети в мастер-станцию и запуск обмена данными (System Q)	G.RLPASET	RLPASET_MD
	GP.RLPASET	RLPASET_P_MD
Передача параметров автоматического обновления (серия "A")	G.RRPA	RRPA_MD
	G.RRPA	RRPA_P_MD
Считывание данных из буферной памяти модуля CC-Link другой станции или из центрального процессора контроллера этой станции	G.RIRD	RIRD_MD
	GP.RIRD	RIRD_P_MD
Запись данных в буферную память модуля CC-Link другой станции или в центральный процессор контроллера этой станции	G.RIWT	RIWT_MD
	GP.RIWT	RIWT_P_MD
Считывание данных из буферной памяти интеллектуальной станции с квитированием установления связи	G.RIRCV	RIRCV_MD
	GP.RIRCV	RIRCV_P_MD
Запись данных в буферную память интеллектуальной станции с квитированием установления связи	G.RISEND	RISEND_MD
	GP.RISEND	RISEND_P_MD
Запись данных в автоматически обновляемую память	G.RITO	RITO_MD
	GP.RITO	RITO_P_MD
Считывание данных из автоматически обновляемой памяти	G.RIFR	RIFR_MD
	GP.RIFR	RIFR_P_MD

### 11.5.1 RLPA (серия "A")

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

Операнды MELSEC A

	Операнды															Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9011	
	битовые					словные (16 бит)					Константы		Указатели		Уровень						
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z						V
n1																	●	●			
d1							●	●	●	●	●										
d2	●	●	●	●	●																

GX IEC Developer

<p>Язык IL для MELSEC</p> <div style="border: 1px solid black; padding: 5px; width: fit-content;"> <p>MELSEC</p> </div> <table style="margin-left: 20px;"> <tr><td>LEDA/LEDB</td><td>RLPA</td></tr> <tr><td>SUB</td><td>n</td></tr> <tr><td>LEDC</td><td>d1</td></tr> <tr><td>LEDC</td><td>d2</td></tr> <tr><td>LEDR</td><td></td></tr> </table>	LEDA/LEDB	RLPA	SUB	n	LEDC	d1	LEDC	d2	LEDR		<p>Релейно-конт. схема (язык LD)</p>	<p>Язык IL по IEC</p> <table style="margin-left: 20px;"> <tr> <td style="border: 1px solid gray; width: 50px; height: 20px;"></td> <td>RLPA_MD</td> <td>n, d1, d2</td> </tr> </table>		RLPA_MD	n, d1, d2
LEDA/LEDB	RLPA														
SUB	n														
LEDC	d1														
LEDC	d2														
LEDR															
	RLPA_MD	n, d1, d2													

GX Developer



Указания по программированию специализированных команд в редакторах MELSEC вы найдете в разделе 3.3 этого руководства.

Переменные

Операнд	Значение		Диапазон	Устанавливает	Тип данных
n	Головной адрес модуля CC-Link на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде 10 <sub>H</sub> .)		от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит
d1	Первый операнд области с параметрами сети				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(d1) + 0	Синхронный режим	<ul style="list-style-type: none"> <li>• [(d1) + 0] = 0: без синхронности</li> <li>• [(d1) + 0] = 1: синхронный режим</li> </ul>	0 или 1	пользователь
	(d1) + 1	Количество подключенных станций	Количество подчиненных станций, подключенных к мастер-станции CC-Link.	от 1 до 64	пользователь
	(d1) + 2	Информация станции (1-я станция)	см. таблицу на следующей странице		
	(d1) + 3	Информация станции (2-я станция)			
	•	•			
	•	•	•		
		Информация станции (последняя станция)	см. таблицу на следующей странице		
		1-я локальная станция	Размер буфера передачи	Количество операндов, обмениваемых между мастер-станцией и локальной или интеллектуальной станцией.	
		локальная станция	Размер прием. буфера		
		станция	Размер автом. обновляемого буфера		
	•	•	•	•	•
•	•	•	•	•	
(d1) + (n - 2)	последняя станция	Размер буфера передачи	как у станции 1	в завис. от используемого модуля	
(d1) + (n - 1)	станция	Размер прием. буфера			
(d1) + n	станция	Размер автом. обновляемого буфера			
d2	Битовый операнд, устанавливаемый на один цикл после выполнения команды RLPA.		0 или 1	система	бит

**Количество операндов для d1:**

Для выбора синхронного режима в (d1) + 0 и указания количества подключенных станций в (d1) + 1 используются по **два** операнда. Для информации каждой станции нужен **один** операнд. Дополнительно, для каждой локальной или интеллектуальной станции должны быть зарезервированы **три** операнда для установления размера буфера.

**Информация станции**

Для каждой станции зарезервировано по одному слову ((d1) + 2, (d1) + 6, (d1) + 10, ...), содержащему подробные данные типа станции:

Значение	Описание	Диапазон
Информация станции	<div style="text-align: center; margin-bottom: 10px;"> </div> <p>0: удал. станц. ввода-вывода                      1: удаленная станция                      2: интеллектуальная станция (включая лок. станции и резервн. мастер-станцию)</p> <p style="margin-left: 100px;">Указание, сколько станций занимает модуль CC-Link</p> <p style="margin-left: 100px;">1: занята 1 станция                      2: заняты 2 станции                      3: заняты 3 станции                      4: заняты 4 станции</p> <p style="margin-left: 100px;">номер станции в диапазоне от 1 до 64</p>	<p>от b0 до b7: 1 – 64 (01<sub>H</sub> – 40<sub>H</sub>)</p> <p>от b8 до b11: 1 – 4</p> <p>от b12 до b15: 0 – 2</p>

**Принцип действия** **Настройка параметров сети CC-Link****RLPA** **Настройка параметров**

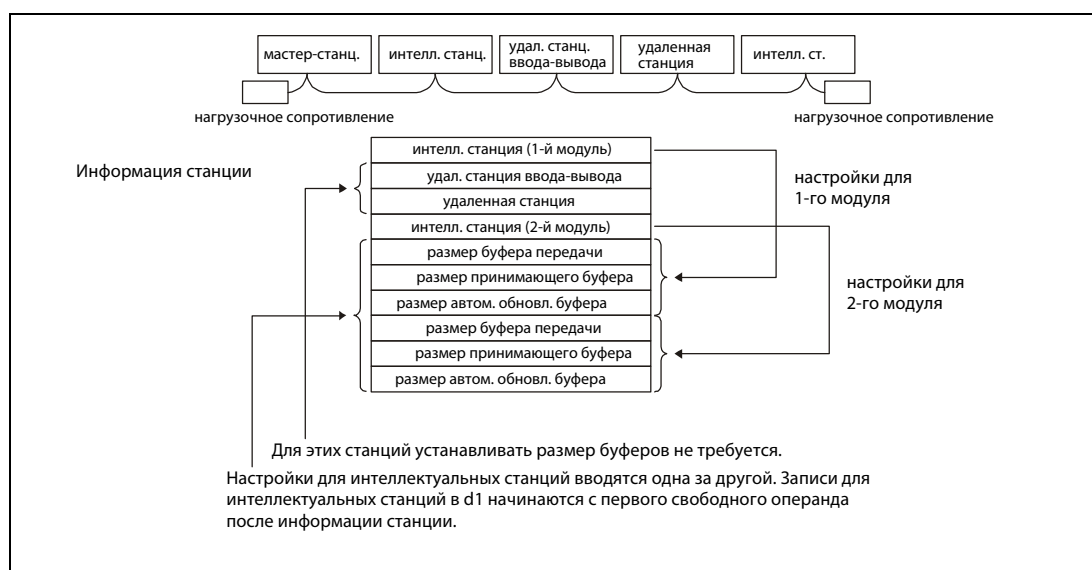
Команда RLPA передает настройки (d1) для сети CC-Link в мастер-станцию (n).

**ПРИМЕЧАНИЕ**

Используйте команду RLPA только для выбора синхронного или несинхронного режима, указания количества подключенных станций, передачи информации станции, указания буфера приема-передачи, а также указания автоматически обновляемого буфера.

Все прочие параметры при выполнении команды RLPA устанавливаются на их стандартные значения. Если параметры одновременно устанавливаются с помощью команд RLPA и TO, то настройки, переданные с помощью команды TO, игнорируются.

Если подчиненная станция (Slave) определена в качестве локальной или интеллектуальной станции, то должен быть установлен размер буфера приема-передачи и автоматически обновляемого буфера в d1. В случае "удаленной станции ввода-вывода" или "удаленной станции" эти данные не нужны. На рисунке ниже показан пример:

**ПРИМЕЧАНИЯ**

Для размера буфера передачи и приема укажите на 7 слов больше, чем это необходимо для обмена данными.

Для автоматически обновляемого буфера требуемый размер присваивает специальный модуль.

Если интеллектуальная станция не поддерживает автоматического обновления или оно в ней не используется, то для размера автоматически обновляемого буфера следует указать значение "0".

Если после установки параметров сети снова выполняется команда RLPA в режиме RUN (с целью изменения параметров сети), то эти новые настройки не используются для коммуникации с подчиненными станциями (Slave).

Лишь после остановки центрального процессора серии "А" (режим STOP/PAUSE) и его повторного перевода в режим RUN обмен данными с подчиненными станциями осуществляется на основе новых параметров.

Выполнение команды RLPA автоматически запускает передачу данных.

Если выполняется команда RLPA, в программе необходимо предусмотреть блокировку с помощью входных сигналов Xn0 (ошибка модуля) и XnF (модуль готов).

**Условия выполнения**

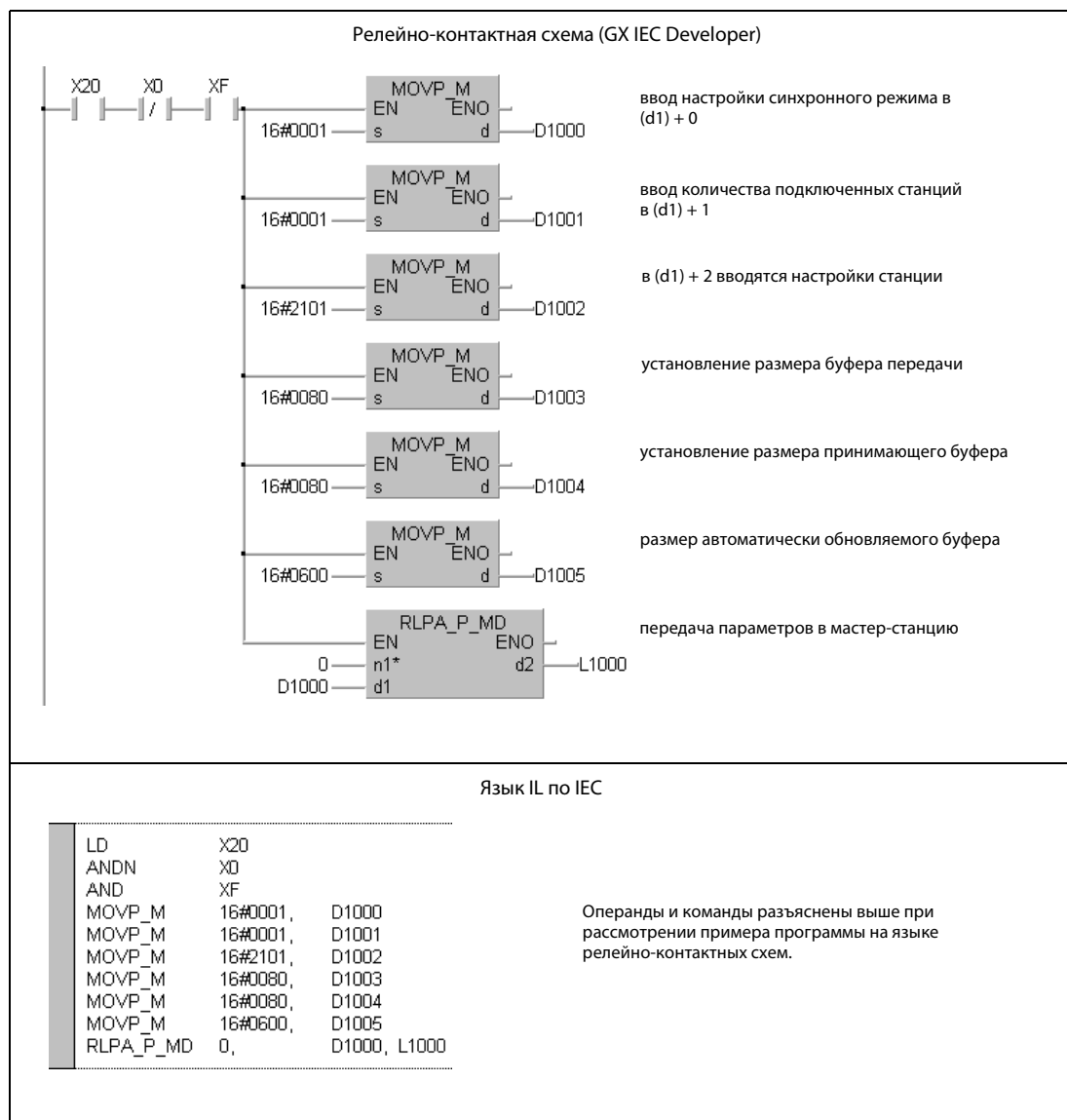
Если команда RLPA запрограммирована в сочетании с командой LEDA, то команда RLPA выполняется до тех пор, пока включено условие выполнения команды LEDA. При применении команды LEDB команды RLPA выполняется только при положительном фронте условия выполнения.

**Пример** RLPA

Эта программа передает параметры сети в мастер-станцию с головным адресом ввода-вывода X/Y000.

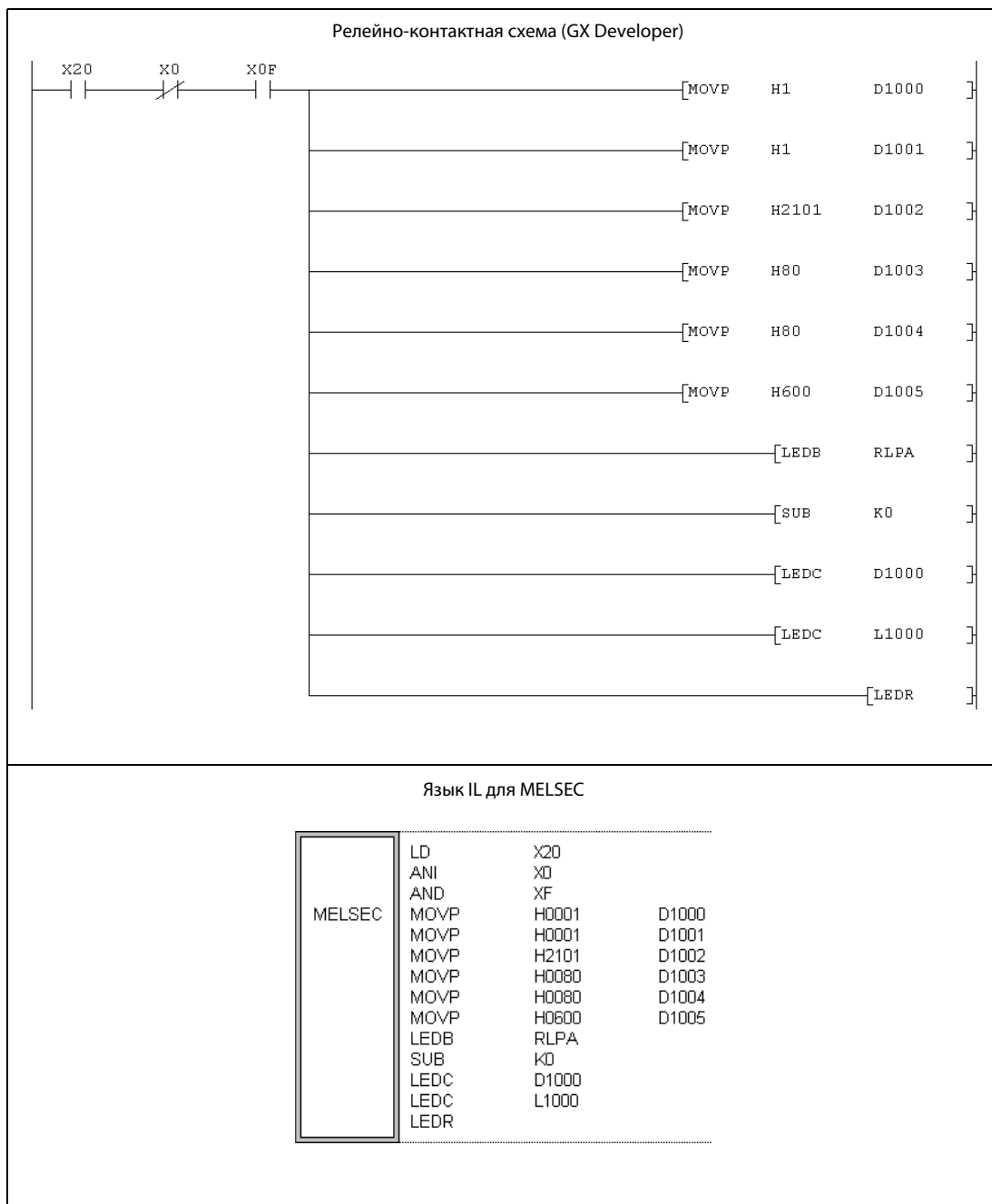
Параметр	Настройка	Значение	Примененный операнд
Синхронный режим	активирован	1	D1000
Количество подключенных станций	1 модуль	1	D1001
Информация станции	Тип станции	интеллектуальная станция	D1002
	Занятые станции	1 станция	
	Номер станции	№ 1	
Размер буфера передачи	128 слов	80 <sub>H</sub>	D1003
Размер принимающего буфера	128 слов	80 <sub>H</sub>	D1004
Размер автоматически обновляемого буфера	960 слов	600 <sub>H</sub>	D1005

- Редакторы по стандарту IEC (На следующей странице этот пример воспроизведен на языке списка инструкций MELSEC и языке LD в среде GX Developer.)





- Список инструкций MELSEC и релейно-контактная схема в среде GX Developer  
 Операнды и команды разъяснены на предыдущей странице при рассмотрении примера программы на языке LD в среде GX IEC Developer.



### 11.5.2 RLPASET (System Q)

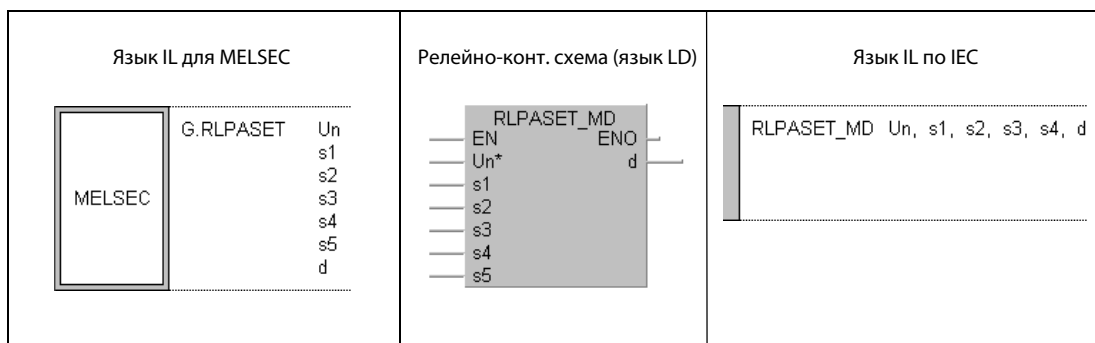
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
					●

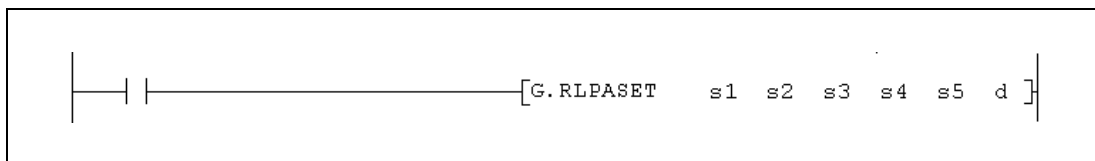
Операнды  
MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□		Спец. модули U□VG□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	—	SM0	
s2	—	●	●	—	—	—	—	—	—		
s3	—	●	●	—	—	—	—	—	—		
s4	—	●	●	—	—	—	—	—	—		
s5	—	●	●	—	—	—	—	—	—		
d	●	●	●	—	—	—	—	—	—		

GX IEC  
Developer



GX  
Developer



## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных		
Un	Головной адрес модуля CC-Link на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде 10 <sub>H</sub> )	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит		
s1	Первый операнд области, содержащей информацию для выполнения команды.					
	Операнд	Значение	Описание	Диапазон	Устанавливает	
	(s1) + 0	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды: 0000 <sub>H</sub> : безошибочная обработка Любое иное значение кроме 0000 <sub>H</sub> : при обработке возникла ошибка.	—	система	
	(s1) + 1	Подтверждение настроек	Первые четыре бита этого операнда указывают, действительны ли настройки, введенные в операндах с s2 по s5: Бит 0 = 1: настройки подчиненных станций (s2) Бит 1 = 1: настройки зарезервир. станций (s3) Бит 2 = 1: настройки игнорирования станций, имеющих неполадки (s4) Бит 3 = 1: настройки буферов приема-передачи и автом. обновл. буферов (s5)  Если настройки не обозначены как действительные, то вместо них принимаются стандартные значения.	от 0 до F	пользователь	
	(s1) + 2	Количество подключенных модулей	Указание подключенных удаленных или локальных модулей (включая зарезервированные станции)	от 1 до 64		Адрес
	(s1) + 3	Количество попыток повторения	Указание числа попыток установления связи с неисправной станцией	от 1 до 7		
	(s1) + 4	Количество модулей с автом. повторным подключением	Указание количества подключенных (удаленных и локальных) модулей, которые после выхода из строя снова автоматически вовлекаются в обмен данными в пределах того же цикла коммуникации.	от 1 до 10		
	(s1) + 5	Режим при останове центрального процессора контроллера	Указание состояния, в которое должна переходить коммуникация при останове центрального процессора контроллера мастер-станции. 0: останов 1: продолжение	0 или 1		
	(s1) + 6	Синхронизация опроса	Выбор между синхронным и асинхронным режимом 0: обмен данными не синхронен с основной программой 1: обмен данными происходит синхронно с выполнением основной программы	0 или 1		
(s1) + 7	Время задержки	Интервал опроса канала связи (единица: 50 мкс)	от 0 до 100			

Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных
s2	Первый операнд области, содержащей настройки подчиненных станций.			
	Операнд	Значение	Описание	Устанавливает
	(s2) + 0	Настройки для станции 1	См. таблицу на странице 94. Выполните настройку для столько модулей, сколько указано в операнде (s1) + 2 (количество подключенных модулей).	пользователь
	(s2) + 1	Настройки для станции 2		
	•	•		
	•	•		
(s2) + 62	Настройки для станции 63			
(s2) + 63	Настройки для станции 64			
s3	Первый операнд области, содержащей настройки зарезервированных станций. Выполните настройку до максимального номера станции, установленного в s2.			
	Операнд	Значение	Описание	Устанавливает
	(s3) + 0	Настройки для станций 1–16	Для каждой станции, которую требуется зарезервировать, устанавливается соответствующий бит (см. таблицу на странице 94). Если модуль занимает более одной станции, то устанавливается бит, соответствующий первому занятому номеру станции.	пользователь
	(s3) + 1	Настройки для станций 17–32		
	(s3) + 2	Настройки для станций 33–48		
(s3) + 3	Настройки для станций 49–64	При предварительной настройке (по умолчанию) ни одна станция не зарезервирована.		
s4	Первый операнд области, содержащей настройки игнорирования неисправных станций. Выполните настройку до максимального номера станции, установленного в s2.			
	Операнд	Значение	Описание	Устанавливает
	(s4) + 0	Настройки для станций 1–16	Для каждой станции, неполадки которой требуется игнорировать, устанавливается соответствующий бит (см. таблицу на странице 94). Если модуль занимает более одной станции, то устанавливается бит, соответствующий первому занятому номеру станции.	пользователь
	(s4) + 1	Настройки для станций 17–32		
	(s4) + 2	Настройки для станций 33–48		
(s4) + 3	Настройки для станций 49–64	Если станция указана как зарезервированная и одновременно требуется игнорировать сообщения этой станции об ошибках, то резервирование имеет более высокий приоритет. При предварительной настройке (по умолчанию) ни одна станция не выбрана.		

## Переменные

Операнд	Значение		Описание	Диапазон	Устанавливает	Тип данных	
s5	Первый операнд области, содержащей настройки размера буферов. Выполните настройку для станций, которые в s2 указаны в качестве локальных или интеллектуальных станций. Начните со станции с наименьшим номером.						
		<b>Операнд</b>	<b>Значение</b>	<b>Описание</b>	<b>Диапазон</b>	<b>Устанавливает</b>	
	(s5) + 0		Размер буфера передачи	Указание размера буфера, используемого для коммуникации между мастер-станцией и локальной или интеллектуальной станцией. Буферы приема-передачи вместе должны иметь размер не больше 4096 слов (1000 <sub>H</sub> ).	0 <sub>H</sub> : без буфера 40 <sub>H</sub> ... 1000 <sub>H</sub> (от 64 до 4096 слов) Предвар. настройка: 40 <sub>H</sub>	пользователь	
	(s5) + 1	1-й модуль	Размер принимающего буфера	Буферная память для коммуникации должна соответствовать объему передаваемых/принимаемых данных плюс по 7 слов.	0 <sub>H</sub> : без буфера 40 <sub>H</sub> ... 1000 <sub>H</sub> (от 64 до 4096 слов) Предвар. настройка: 40 <sub>H</sub>		
	(s5) + 2		Размер автоматич. обновляемого буфера	Количество операндов автоматически обновляемого буфера, используемых для коммуникации между мастер-станцией и локальной/интеллектуальной станцией. Размер автоматически обновляемого буфера должен соответствовать количеству данных, которыми предусматривается обмениваться с интеллектуальными станциями.	0 <sub>H</sub> : без буфера 80 <sub>H</sub> ... 1000 <sub>H</sub> (от 128 до 4096 слов) Предвар. настройка: 80 <sub>H</sub>		
	•		•	•	•		•
	(s5) + 75	26 модуль	Размер буфера передачи	как у 1-го модуля			
	(s5) + 76		Размер принимающего буфера				
	(s5) + 77		Размер автоматич. обновляемого буфера				
	d	Битовый операнд, устанавливаемый на один цикл после выполнения команды RLPASET. С помощью операнда (d) + 1 выводится ошибка при выполнении.					
		<b>Операнд</b>	<b>Значение</b>	<b>Описание</b>	<b>Диапазон</b>		<b>Устанавливает</b>
(d) + 0		Команда выполнена	Показывает завершение команды RLPASET. ВКЛ.: команда выполнена ВыКЛ.: команда не выполнена	0 или 1	система		
(d) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды RLPASET. ВКЛ.: команда выполнена с ошибкой ВыКЛ.: команда выполнена без ошибок	0 или 1				
					бит		

### Настройки подчиненных станций (Slave)

Для каждой станции в параметрах зарезервировано по одному слову (от (s2) + 0 до (s2) + 63), содержащему подробную информацию о типе станции:

Значение	Описание	Диапазон
Настройки отдельных станций	<div style="text-align: center;"> </div> <p>0: удал. станц. ввода-вывода                      1: удаленная станция                      2: интеллектуальная станция (включая лок. станции и резервн. мастер-станцию)</p> <p style="text-align: center;">Указание, сколько станций занимает модуль CC-Link:                      1: занята 1 станция                      2: заняты 2 станции                      3: заняты 3 станции                      4: заняты 4 станции</p>	от b0 до b7: 1-64 (01 <sub>H</sub> -40 <sub>H</sub> ) от b8 до b11: 1-4 от b12 до b15: 0-2

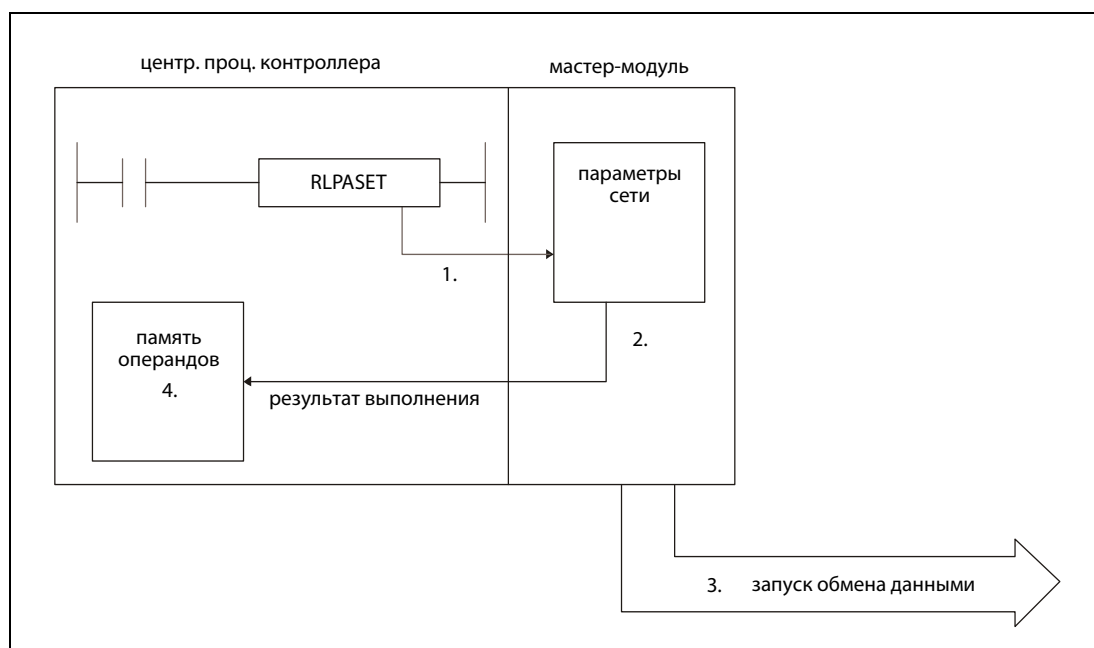
Предварительными настройками (настройками по умолчанию) операндов с (s2) + 0 по (s2) + 63 являются настройки от "0101<sub>H</sub>" до "0140<sub>H</sub>". (Номер станции от 1 до 64, занимает одну станцию, удаленная станция ввода-вывода)

### Указание номера станции в s3 и s4

Каждый бит в четырех словных операндах, используемых для s3 и s4, представляет одну станцию:

Операнд	Бит															
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
(s3) + 0 (s4) + 0	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
(s3) + 1 (s4) + 1	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
(s3) + 2 (s4) + 2	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33
(s3) + 3 (s4) + 3	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49

Числа с 1 по 64 означают номера станций. Станция выбирается путем установки соответствующего бита.

**Принцип действия****Настройка параметров сети CC-Link  
RLPASET      Настройка параметров**

1. Команда RLPASET передает параметры сетевой коммуникации (от s1 до s5) в мастер-станцию (Un).
2. Главный модуль (мастер) проверяет настроенные параметры.
3. Если ошибок в них нет, запускается обмен данными.
4. Устанавливается операнд d.

Одновременно возможно выполнение только одной команды RLPASET.

**Занимаемые операнды**

Команда RLPASET занимает следующее количество операндов:

- s1: 8 словных операндов
- s2: 64 словных операнда
- s3: 4 словных операнда
- s4: 4 словных операнда
- s5: 78 словных операндов

При программировании учитывайте необходимый размер областей для хранения данных, указанных в s1...s5.

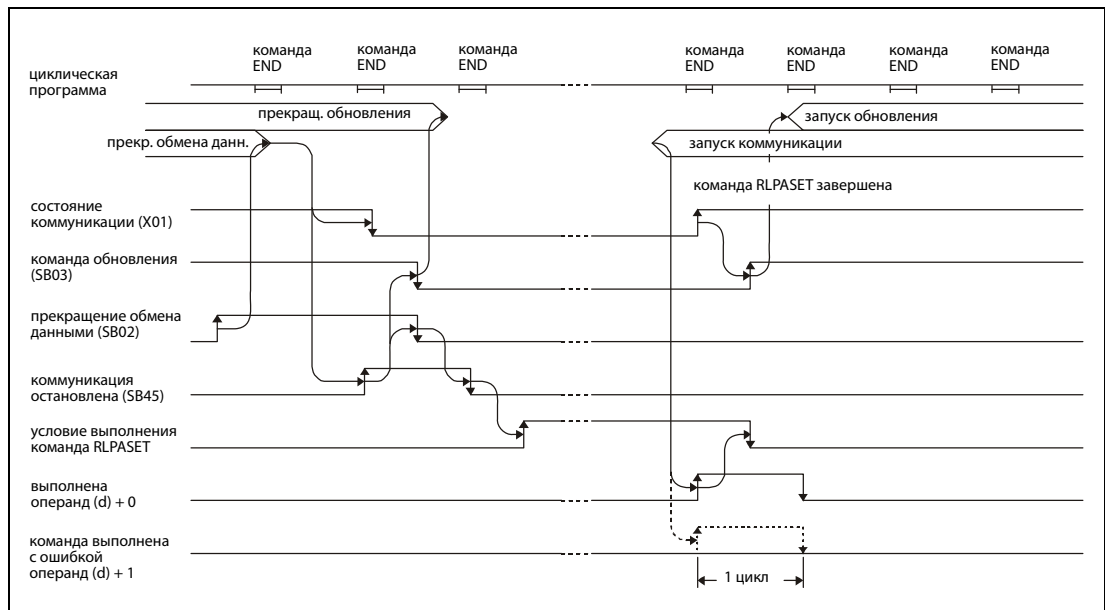
Пример:

К мастер-станции подключены четыре подчиненных станции. Примененный в контроллере процессор Q02 имеет регистры данных с D0 по D12287. Если для s2 указать регистр данных D12284, так как имеются только 4 подчиненных устройства, то при выполнении команды RLPASET возникнет сообщение об ошибке с кодом 4101, так как центральный процессор всегда проверяет область адресов для 64 станций (в этом случае – регистры с D12284 по D12347), и происходит превышение имеющегося диапазона адресов.

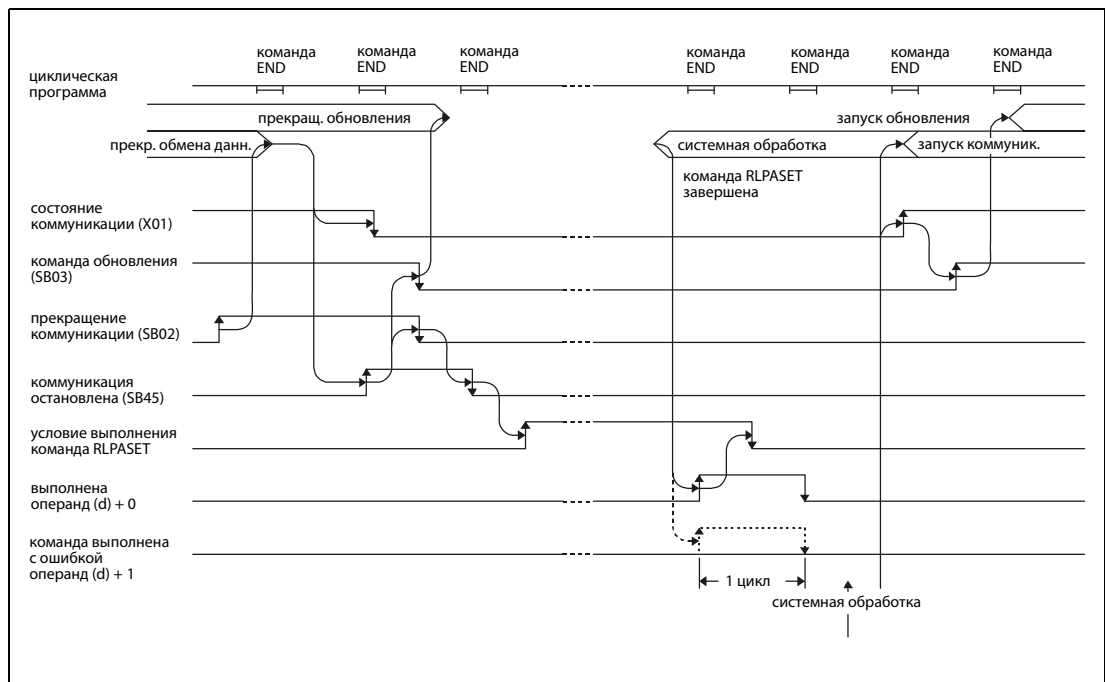
Операнды, указанные в (d) + 0 и (d) + 1, показывают, выполнена ли команда и возникла ли при этом ошибка.

- Битовый операнд (d) + 0 устанавливается при выполнении команды END того цикла, в котором была завершена команда RLPASET. При обработке следующей команды END этот битовый операнд снова сбрасывается.
- Битовый операнд (d) + 1 показывает ошибку при выполнении команды RLPASET. При безошибочном выполнении этот битовый операнд остается сброшенным. Если имеется ошибка, то операнд (d) + 1 устанавливается при выполнении команды END того цикла, в котором была завершена команда RLPASET. При обработке следующей команды END этот операнд снова сбрасывается.

На рисунке ниже показана диаграмма изменения сигналов при выполнении команды RLPASET и безошибочной работе станций:



На следующей иллюстрации показана диаграмма выполнения команды RLPASET при наличии ошибок в станциях:





**Источники ошибок**

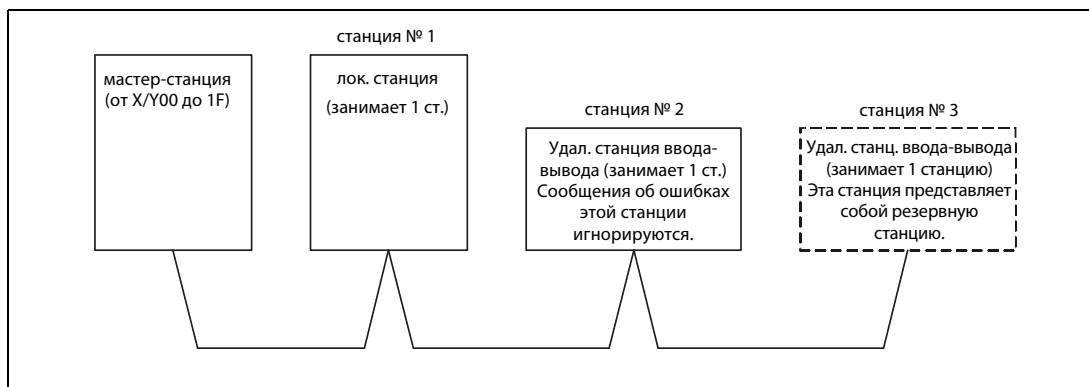
При следующих событиях при выполнении команды RLPASET устанавливается флаг ошибки SMO и в специальный регистр SD0 записывается код ошибки:

- Указанный в Un модуль не является специальным модулем (код ошибки: 2112).
- Выполняемая команда не поддерживается (код ошибки: 4002).
- В команде указано недействительное количество операндов (код ошибки: 4003).
- В команде указан недопустимый операнд (код ошибки: 4004).
- Команда содержит недопустимые данные (код ошибки: 4100).
- Указанные операнды превышают допустимый диапазон (код ошибки: 4101).
- Сохраненные данные или константы, переданные с помощью этой команды, превышают допустимый диапазон (код ошибки: 4101).

**Пример**

RLPASET

Эта программа передает параметры сетевой коммуникации для мастер-станции с головным адресом ввода-вывода X/Y000. Сеть CC-Link состоит из трех подчиненных станций:



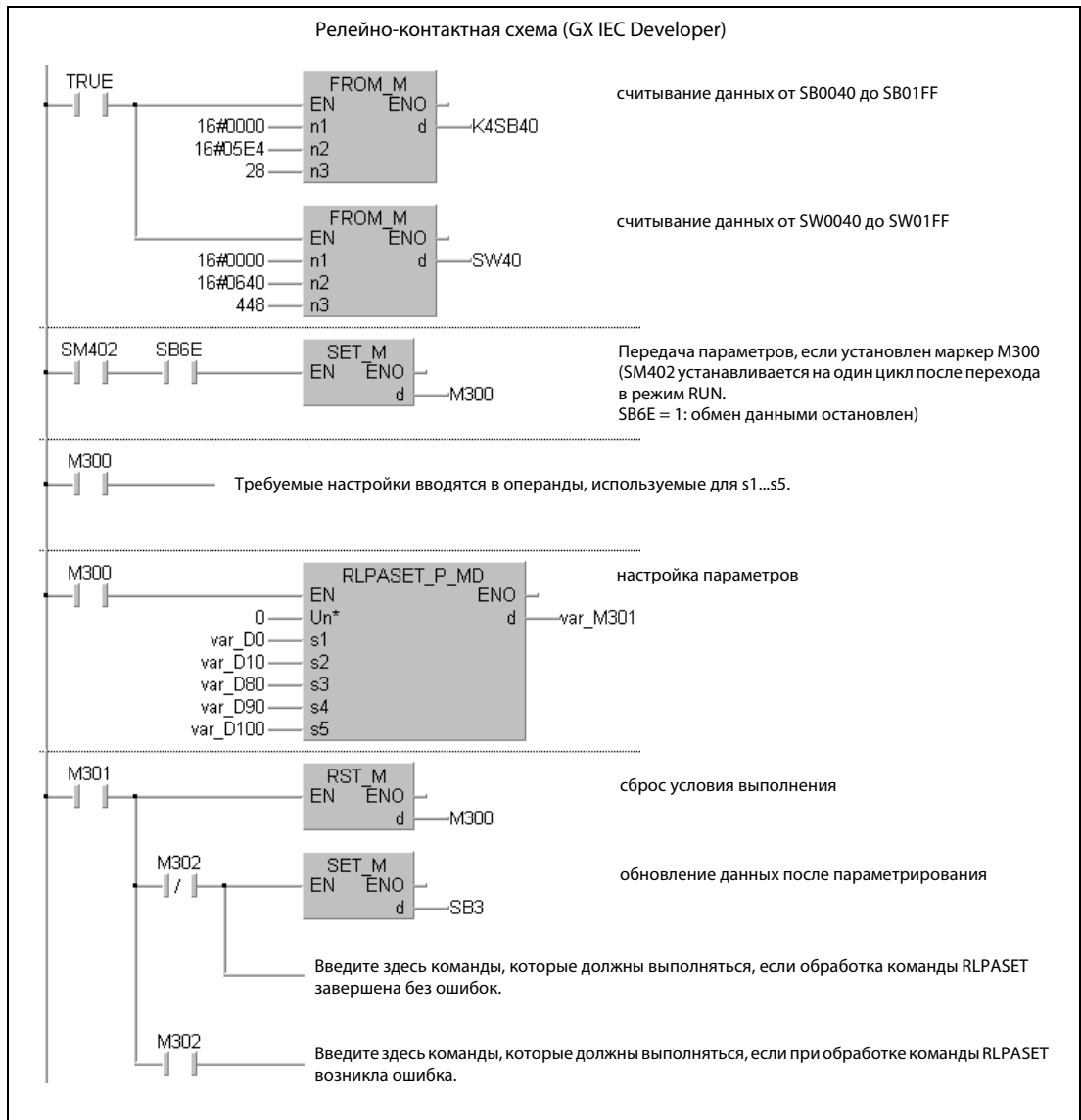
Переменные от s1 до s5 команды RLPASET содержат следующие значения:

Параметр		Настройка	Записанное значение	Операнд для сохранения	
Данные для выполнения команды	(s1) + 1	Подтверждение настроек	Действительны все настройки	15	D1
	(s1) + 2	Количество подключенных модулей	3 подчиненных модуля	3	D2
	(s1) + 3	Количество попыток повторения	3 попытки	3	D3
	(s1) + 4	Количество модулей с автомат. повторным подключением	1 модуль	1	D4
	(s1) + 5	Режим при останове центрального процессора контроллера	прекращение обмена данными	0	D5
	(s1) + 6	Синхронизация опроса	асинхронный режим	0	D6
	(s1) + 7	Время задержки	0 мкс	0	D7
Настройки для подчиненных станций	(s2) + 0	Настройки для первой станции	локальная станция, занимает одну станцию, № станции: 1	2101 <sub>H</sub>	D10
	(s2) + 1	Настройки для второй станции	удаленная станция ввода-вывода, занимает одну станцию, № станции: 2	102 <sub>H</sub>	D11
	(s2) + 2	Настройки для третьей станции	удаленная станция ввода-вывода, занимает одну станцию, № станции: 3	103 <sub>H</sub>	D12
Зарезервированные станции	(s3) + 0	Настройки зарезервированных станций	зарезервировать станцию 3 (установлен бит 2)	4	D80
	(s3) + 1			0	D81
	(s3) + 2			0	D82
	(s3) + 3			0	D83
Игнорирование неисправных станций	(s4) + 0	Настройки для игнорирования сообщений об ошибках станций, имеющих неполадки	игнорировать сообщения об ошибках станции 2 (установлен бит 1)	2	D90
	(s4) + 1			0	D91
	(s4) + 2			0	D92
	(s4) + 3			0	D93

Параметр		Настройка	Записанное значение	Операнд для сохранения	
Настройка размера буфера	(s5) + 0	Буфер передачи первой локальной станции (станция № 1)	100 слов	64 <sub>H</sub>	D100
	(s5) + 1	Приемный буфер первой локальной станции (станция № 1)	100 слов	64 <sub>H</sub>	D101
	(s5) + 2	Автоматически обновляемый буфер первой локальной станции (станция № 1)	не используется	0 <sub>H</sub>	D102

Перед выполнением команды RLPASET необходимо записать значения параметров в регистры данных D1...D102 в соответствии с таблицей.

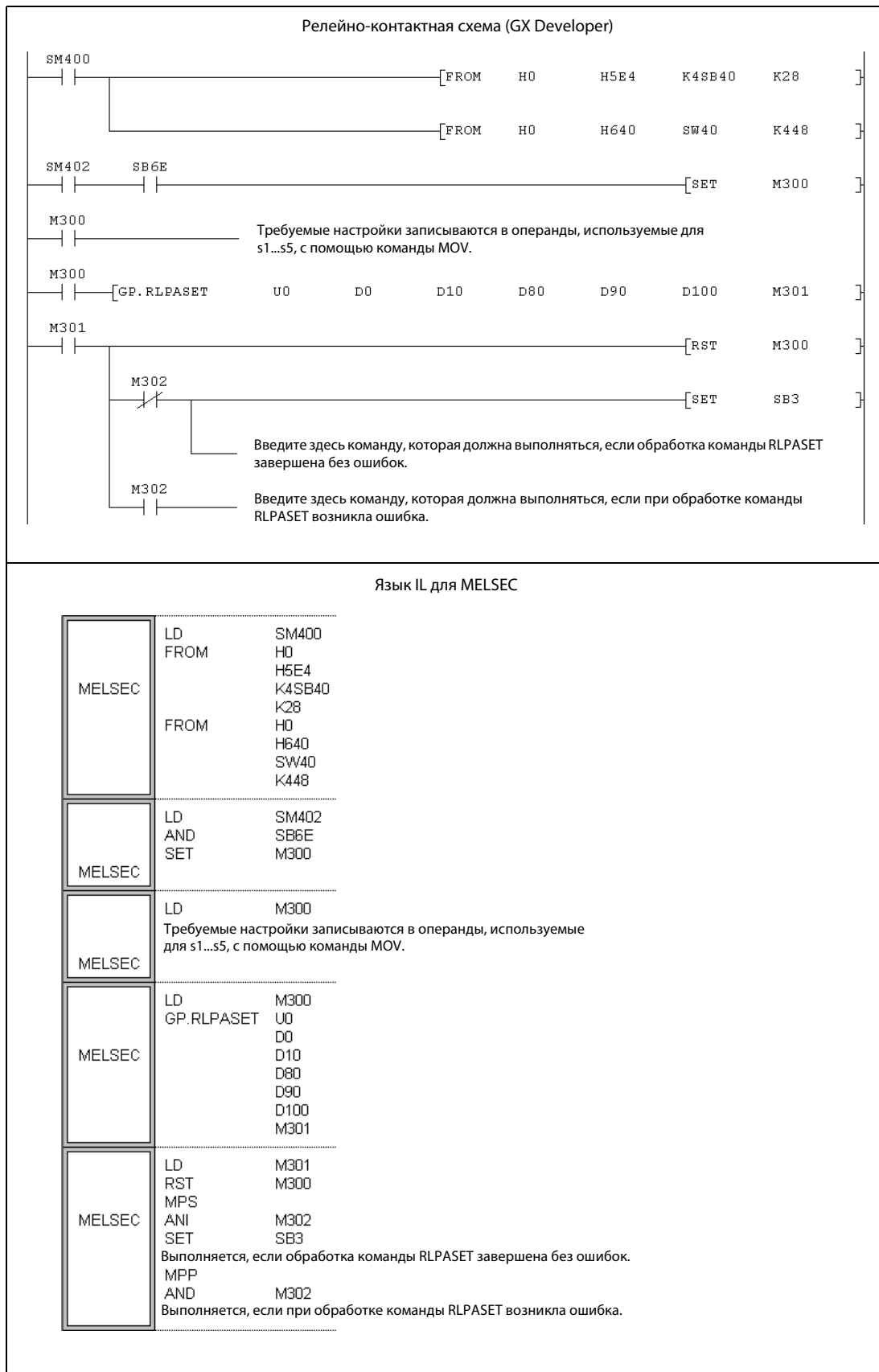
- Редакторы по стандарту IEC (на следующих страницах этот пример отображен на языке IL для MELSEC и языке LD в среде GX Developer)



Язык IL по IEC					
LD	TRUE				
FROM_M	16#0000,	16#05E4,	28,	K4SB40	
FROM_M	16#0000,	16#0640,	448,	SW40	
.....					
LD	SM402				
AND	SB6E				Операнды и команды разъяснены при рассмотрении
SET_M	M300				примера программы на языке LD.
.....					
LD	M300				
	Требуемые настройки записываются в операнды, используемые для s1...s5, с помощью команды MOV.				
.....					
LD	M300				
RLPASET_P_MD	0,	var_D0,	var_D10,	var_D80,	var_D90,
					var_D100,
					var_M301
.....					
LD	M301				
RST_M	M300				
LD	M301				
ANDN	M302				
SET_M	SB3				
	Выполняется, если обработка команды RLPASET завершена без ошибок.				
LD	M301				
AND	M302				
	Выполняется, если при обработке команды RLPASET возникла ошибка.				

**ПРИМЕЧАНИЕ** *В редакторах по стандарту IEC среды GX IEC Developer переменные должны быть объявлены в заголовке программного компонента (POU). Если переменные не были объявлены, то при проверке или компиляции программы возникает сообщение об ошибке. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.*

- Список инструкций MELSEC и релейно-контактная схема в среде GX Developer  
Операнды и команды разъяснены при рассмотрении примера программы на языке LD в среде GX IEC Developer.



### 11.5.3 RRPA (серия "A")

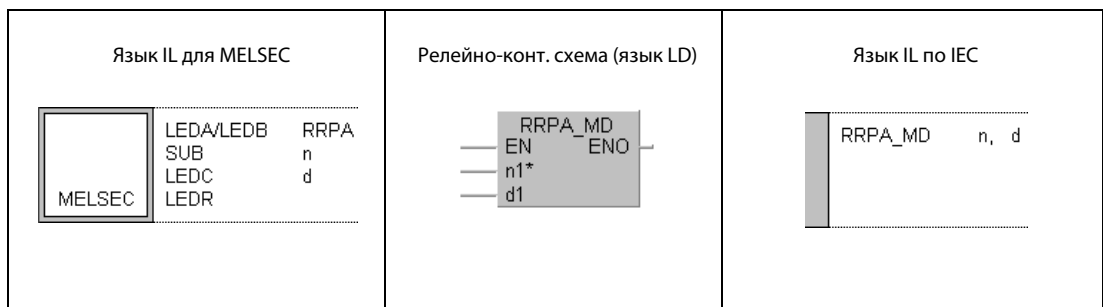
Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

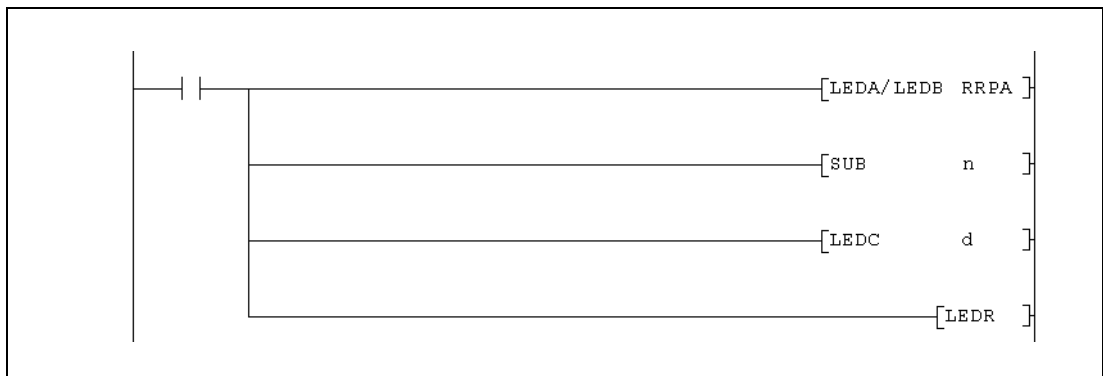
Операнды  
MELSEC A

Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9011			
битовые						словные (16 бит)						Константы		Указатели							Уровень		
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						K	H (16#)	P
n																●	●						
d						●	●	●	●	●													●

GX IEC  
Developer



GX  
Developer



Указания по программированию специализированных команд в редакторах MELSEC вы найдете в разделе 3.3 этого руководства.

## Переменные

Операнд	Значение		Диапазон	Устанавливает	Тип данных	
n	Головной адрес модуля CC-Link на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде 10 <sub>н</sub> )		от 0 до FE <sub>н</sub>	пользователь	BIN, 16 бит	
d	Первый операнд области с параметрами сети					
	Операнд	Значение	Описание	Устанавливает	BIN, 16 бит	
	d + 0	R X	Начальный адрес	Начальный адрес децентрализованных входов (RX) в мастер-модуле или локальном модуле		система
	d + 1		Операнды в центральный процессоре	Закодированное указание операндов, которые автоматически обновляются в центральном процессоре (см. ниже).		пользователь
	d + 2		Начальный адрес в ЦП	Первый адрес области операндов в центральном процессоре*		
	d + 3		Количество операндов	Количество операндов, обмениваемых между модулем CC-Link и центральным процессором*		
	d + 4	R Y	Начальный адрес	Начальный адрес децентрализованных выходов (RY) в мастер-модуле или локальном модуле		пользователь
	d + 5		Операнды в центральный процессоре	Закодированное указание операндов, которые автоматически обновляются в центральном процессоре (см. ниже).		
	d + 6		Начальный адрес в ЦП	Первый адрес области операндов в центральном процессоре*		
	d + 7		Количество операндов	Количество операндов, обмениваемых между модулем CC-Link и центральным процессором*		
	d + 8	R W	Начальный адрес	Начальный адрес децентрализованных регистров (RW) в мастер-модуле или локальном модуле		RW: система
	d + 9		Операнды в центральный процессоре	Закодированное указание операндов, которые автоматически обновляются в центральном процессоре (см. ниже).		пользователь
	d + 10		Начальный адрес в ЦП	Первый адрес области операндов в центральном процессоре*		
	d + 11		Количество операндов	Количество операндов, обмениваемых между модулем CC-Link и центральным процессором*		
	d + 12	S B	Начальный адрес	Начальный адрес специальных маркеров связи (SB) в мастер-модуле или локальном модуле		система
	d + 13		Операнды в центральный процессоре	Закодированное указание операндов, которые автоматически обновляются в центральном процессоре (см. ниже).		пользователь
	d + 14		Начальный адрес в ЦП	Первый адрес области операндов в центральном процессоре*		
	d + 15		Количество операндов	Количество операндов, обмениваемых между модулем CC-Link и центральным процессором*		
	d + 16	S W	Начальный адрес	Начальный адрес специальных регистров связи (SW) в мастер-модуле или локальном модуле		система
	d + 17		Операнды в центральный процессоре	Закодированное указание операндов, которые автоматически обновляются в центральном процессоре (см. ниже).		пользователь
d + 18	Начальный адрес в ЦП		Первый адрес области операндов в центральном процессоре*			
d + 19	Количество операндов		Количество операндов, обмениваемых между модулем CC-Link и центральным процессором*			

\* В случае битовых операндов (X, Y, M, B) в качестве начального адреса должен быть указан либо 0, либо число, кратное 16. (Если указан 0, автоматическое обновление соотв. операндов не происходит.) Если не соблюсти это требование, возникнет сообщение об ошибке.

В  $d + 5$ ,  $d + 13$  и т. п. устанавливается, какие операнды в центральном процессоре соответствуют операндам модуля CC-Link. Например, для отображения состояний децентрализованных входов (RX) могут использоваться маркеры (M).

Код	Операнд	Код	Операнд
0	—	5	T
1	X	6	C
2	Y	7	D
3	M	8	W
4	B	9	R

**Принцип действия**

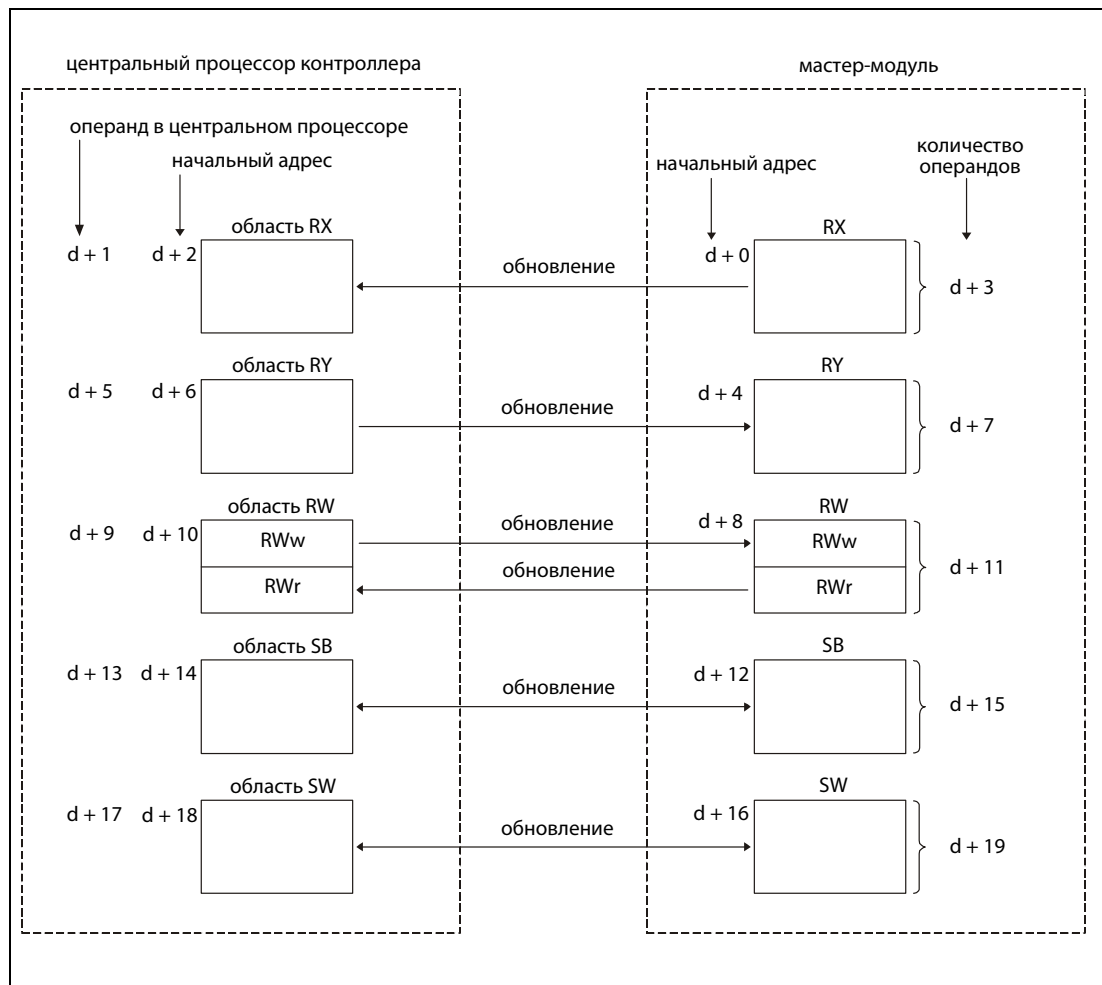
**Настройка параметров для автоматического обновления**

**RRPA Настройка параметров**

Команда RRPA указывает операнды и количество операндов, состояния которых автоматически передаются между центральным процессором контроллера и мастер-модулем или локальным модулем.

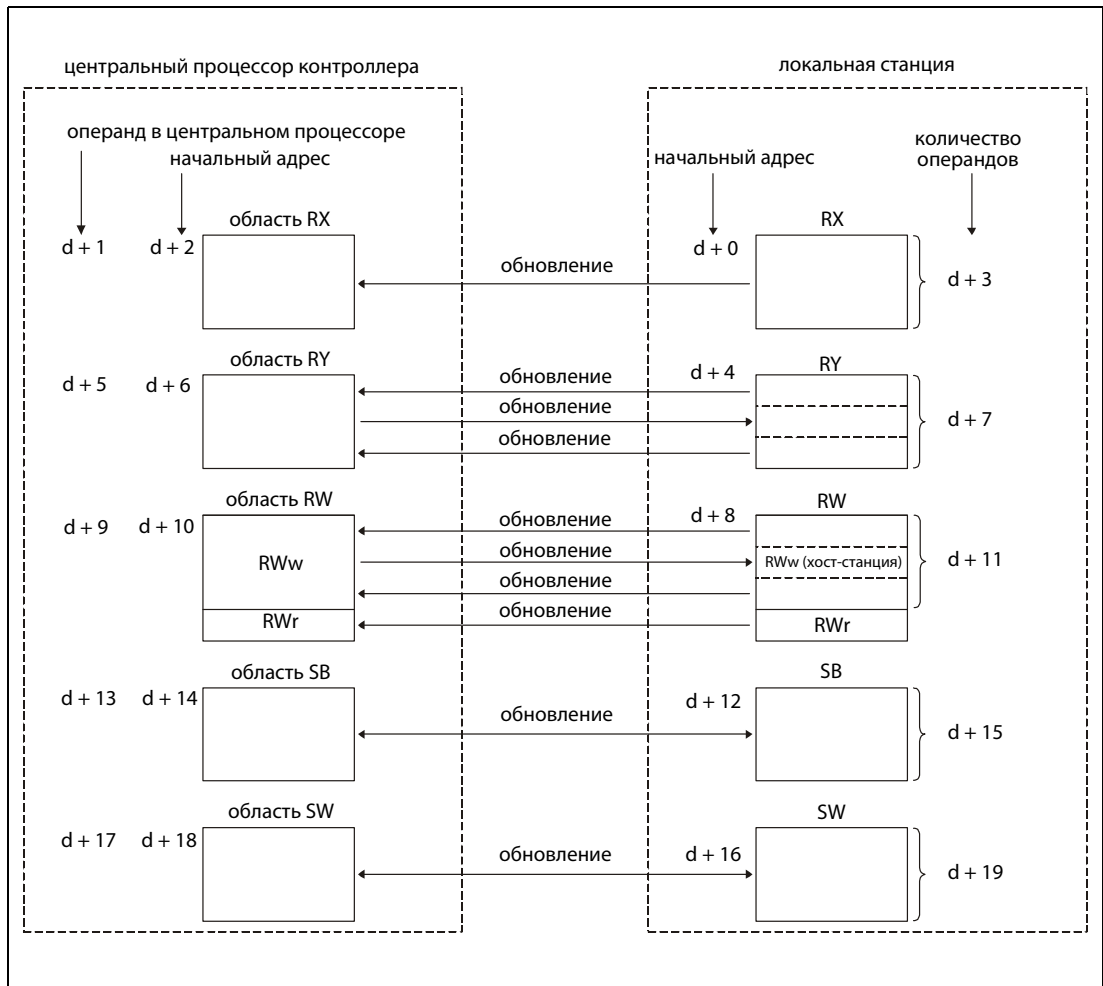
Если для обмена данными между центральным процессором и мастер-модулем или локальным модулем используются команды FROM и TO, применять команду RRPA не требуется.

- Коммуникация между центральным процессором контроллера и мастер-станцией:





- Коммуникация между центральным процессором контроллера и локальной станцией:

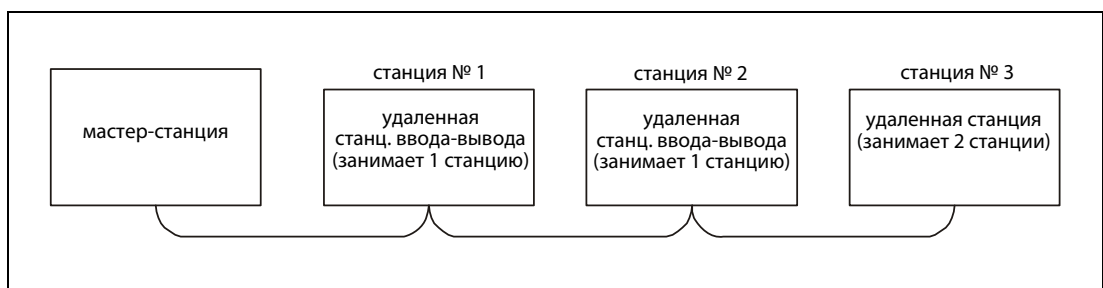


При выполнении команды RRPA параметры автоматического обновления вводятся в центральный процессор, а затем на основе этих настроек происходит обмен данными между центральным процессором и мастер-модулем или локальным модулем.

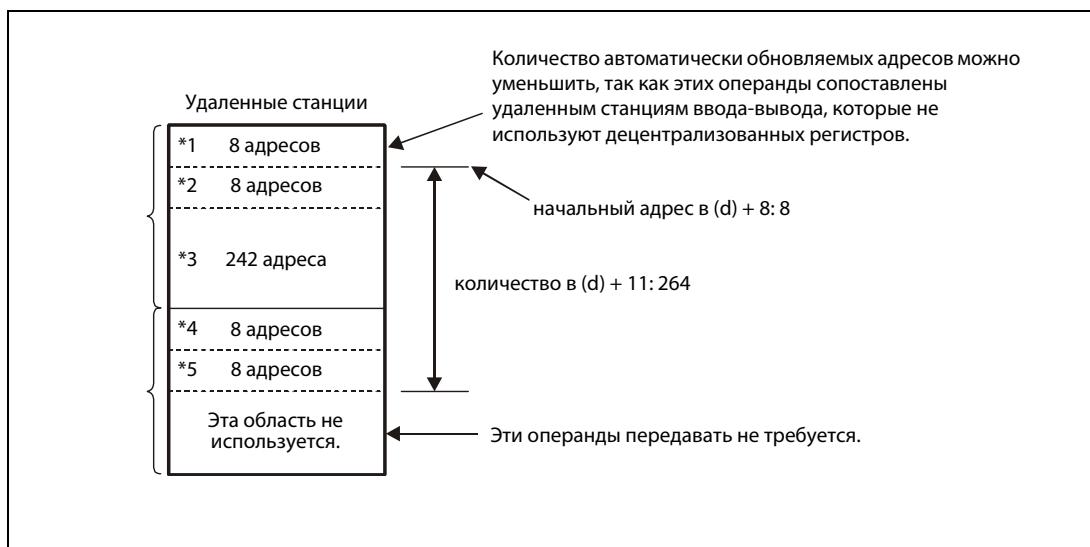
Команду RRPA требуется выполнить только один раз. Если для одного модуля запрограммированы несколько команд RRPA, то действительны только настройки, установленные первой командой. Для изменения параметров выполняется команда RRPA с новыми настройками. Чтобы новые параметры начали действовать, необходимо переключить центральный процессор контроллера в режим STOP/PAUSE, а затем снова в режим RUN.

Чтобы обновлять все децентрализованные регистры (RWw и RWr), введите в качестве начального адреса в d + 8 значение 0, а в d + 11 укажите количество "512".

**ПРИМЕЧАНИЕ** Следующий пример разъясняет порядок действий при обновлении децентрализованных регистров:



Даже если подключено менее 64 станций, в децентрализованных регистрах для RWr занимают все 256 слов (64 станции). Поэтому область для RWr начинается после 256 адресов RWw.



- \*1: Область RWw (8 операндов) станций 1 и 2 (удаленные станции ввода-вывода)  
 \*2: Область RWw (8 операндов) станции 3 (удаленная станция)  
 \*3: 242 операнда области RWw автоматически занимает система  
 \*4: Область RWr (8 операндов) станций 1 и 2 (удаленные станции ввода-вывода)  
 \*5: Область RWr (8 операндов) станции 3 (удаленная станция)

#### Обновление операндов SB и SW:

- Сопоставьте специальным маркерам связи (SB) и специальным регистрам (SW) операнды в центральном процессоре контроллера. При этом соблюдайте начальные адреса областей: операнды от SB0000 до SB003F передаются от центрального процессора контроллера в мастер-модуль, а операнды от SB0040 до SB00FF – от мастер-модуля в центральный процессор.
- В качестве обновляемых операндов для SB и SW нельзя указывать регистры файлов. Если для SB или SW указаны регистры файлов и эта настройка передана в центральный процессор, то распознается ошибка кода команды и центральный процессор останавливается.
- Операнды, сопоставленные специальным маркерам связи (SB) и специальным регистрам (SW), не должны быть определены в качестве области фиксации. В противном случае при включении электропитания или сбросе центрального процессора не будет обеспечена нормальная работа (из-за неопределенности данных).
- Обновляемые области SB и SW, установленные с помощью команды RRPА при включении напряжения питания, не могут быть изменены.

#### Условия выполнения

Если команда RRPА запрограммирована в сочетании с командой LEDA, то команда RRPА выполняется до тех пор, пока условие выполнения команды LEDA включено. При применении команды LEDB команда RRPА выполняется только при положительном фронте условия выполнения.

#### Источники ошибок

При следующих событиях распознается ошибка обработки, устанавливается флаг ошибки M9011 и в регистр D908 записывается код ошибки "50". (В процессорах AnU записывается код ошибки "503" в регистр D9001, а в процессорах AnSH - код ошибки "503" в регистр D9092.)

- Код операнда равен 0 или числу вне диапазона 1...9.
- В случае битового операнда начальный адрес не равен 0 и не кратен 16.
- Количество обновляемых адресов не кратно 16.

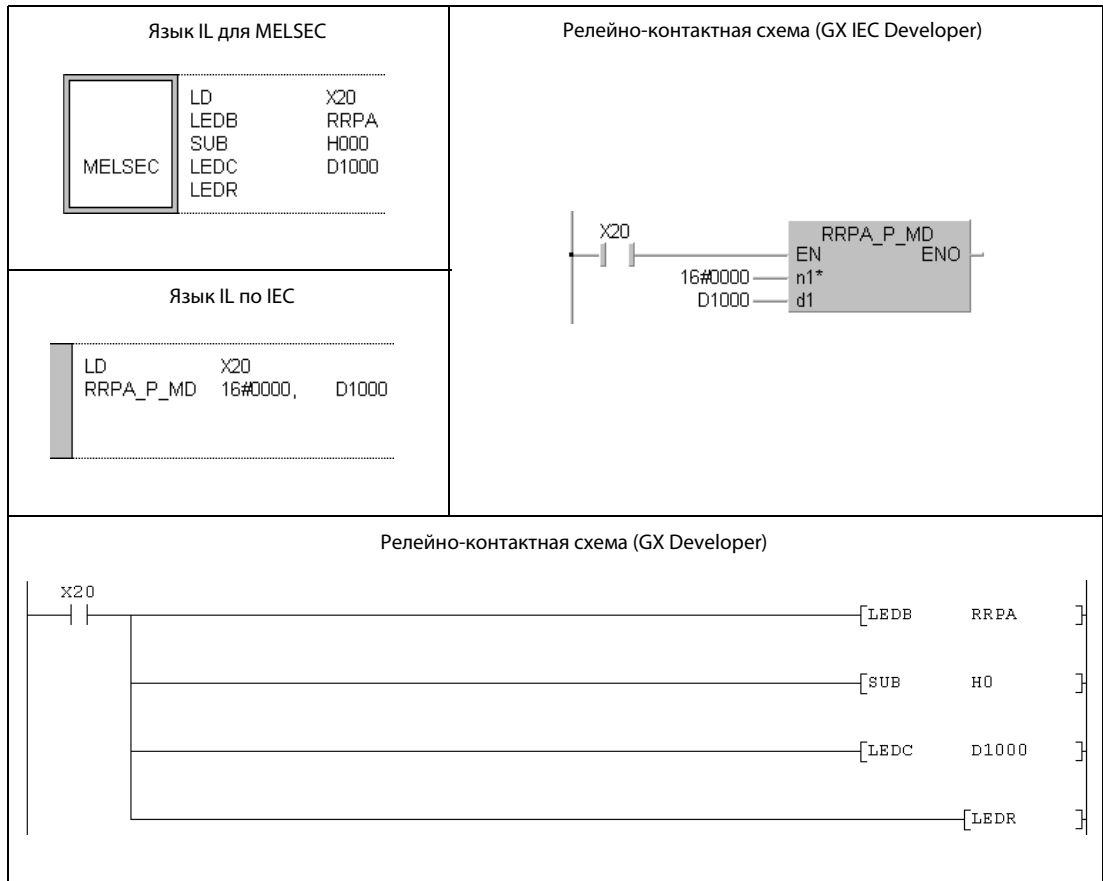
**Пример**

## RRPA

Эта программа передает параметры автоматического обновления, сохраненные, начиная с регистра файлов D1000, в мастер-станции с головным адресом ввода-вывода X/Y000.

Параметр		Настройка	Записанное значение	Операнд для сохранения
RX	Начальный адрес	0	0 <sub>H</sub>	D1000
	Операнды в центральном процессоре	X (код: 1)	1 <sub>H</sub>	D1001
	Начальный адрес в центральном процессоре	XA0	A0 <sub>H</sub>	D1002
	Количество операндов	32	32	D1003
RY	Начальный адрес	0	0 <sub>H</sub>	D1004
	Операнды в центральном процессоре	Y (код: 2)	2 <sub>H</sub>	D1005
	Начальный адрес в центральном процессоре	YA0	A0 <sub>H</sub>	D1006
	Количество операндов	48	48	D1007
RW	Начальный адрес	0	0 <sub>H</sub>	D1008
	Операнды в центральном процессоре	D (код: 7)	7 <sub>H</sub>	D1009
	Начальный адрес в центральном процессоре	D160	160	D1010
	Количество операндов	272	272	D1011
SB	Начальный адрес	0	0 <sub>H</sub>	D1012
	Операнды в центральном процессоре	M (код: 3)	3 <sub>H</sub>	D1013
	Начальный адрес в центральном процессоре	M160	160	D1014
	Количество операндов	256	256	D1015
SW	Начальный адрес	0	0 <sub>H</sub>	D1016
	Операнды в центральном процессоре	W (код: 8)	8 <sub>H</sub>	D1017
	Начальный адрес в центральном процессоре	WA0	A0 <sub>H</sub>	D1018
	Количество операндов	256	256	D1019

Перед выполнением команды RRPA значения параметров должны быть записаны в регистры данных с D1000 по D1019 в соответствии с таблицей.



### 11.5.4 RIRD (серия "A")

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

Операнды MELSEC A

Операнды																Длина блока	Шагов	Индекс	Флаг переноса	Флаг ошибки	
битовые						словные (16 бит)						Константы		Указатели							Уровень
X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						
n1																●	●				
n2																●	●				
d1							●	●	●	●	●										
d2	●	●	●	●	●																

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных
n1	Головной адрес мастер-станции или локальной станции CC-Link (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде 10H.)	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит
n2	Номер станции, из которой требуется считать данные. Диапазон: если команда RIRD выполняется в мастер-станции: от 1 до 64; если команда RIRD выполняется в локальной станции: от 0 до 64.	см. разьяснение рядом	пользователь	BIN, 16 бит

## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных		
d1	Первый операнд области, содержащей информацию для выполнения команды и сохранения считанных данных.				BIN, 16 бит	
	Операнд	Значение	Описание	Диапазон		Устанавливает
	(d1) + 0	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды: 0000 <sub>H</sub> : безошибочная обработка Любое иное значение кроме 0000 <sub>H</sub> : при обработке возникла ошибка. Более подробные указания, относящиеся к кодам ошибок, имеются в руководстве по эксплуатации модуля CC-Link.	—		система
	(d1) + 1	Длина данных	Укажите здесь, сколько данных (слов) требуется считать. Длина данных зависит от типа центрального процессора станции, из которой происходит считывание: Серии AnU, QnA, System Q: макс. 480 слов Все прочие центральные процессоры: макс. 32 слова	от 1 до 480 от 1 до 32		пользователь
	(d1) + 2	Код доступа	<ul style="list-style-type: none"> <li>В случае мастер-модуля с версией программного обеспечения от "A" до "H" При указании 0004<sub>H</sub> считывание происходит из буферной памяти интеллектуальной станции. Если требуется считывать из буферной памяти локальной станции, введите значение 2004<sub>H</sub>.</li> </ul>	0004 <sub>H</sub> или 2004 <sub>H</sub>		
	(d1) + 3	Начальный адрес	<ul style="list-style-type: none"> <li>В случае мастер-модуля с версией программного обеспечения от "A" до "H" Начальный адрес в буферной памяти</li> </ul>	в завис. от станции, к которой осуществляется доступ		
			<ul style="list-style-type: none"> <li>В случае мастер-модуля с версией программного обеспечения, начиная с "J" Начальный адрес в буферной памяти или первый адрес операнда.</li> </ul>			
от (d1) + 4 до (d1) + n	Область памяти для считанных данных	Размер этой области определяется количеством данных, указанных в (d1) + 1.	—	система		

## Переменные

Операнд	Значение			Диапазон	Устанавливает	Тип данных
d2	Битовый операнд, устанавливаемый на один цикл после выполнения команды RIRD. По адресу (d2) + 1 выводится ошибка при выполнении.					бит
	Операнд	Значение	Описание	Диапазон	Устанавливает	
	(d2) + 0	Команда выполнена	Показывает завершение команды. ВКЛ.: команда выполнена ВыКЛ.: команда не выполнена	—	система	
(d2) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды. ВКЛ.: команда выполнена с ошибкой ВыКЛ.: команда выполнена без ошибок	—			

Начиная с версии "J" программного обеспечения главного модуля, считываемые данные указываются с помощью двух кодов в (d1) + 2. Они указывают, из какой части буферной памяти происходит считывание или какие операнды центрального процессора определяются:

- Доступ к буферной памяти модуля CC-Link (код доступа 04<sub>H</sub>)

Цель доступа	Код операнда	
Буферная память в интеллектуальных станциях	00 <sub>H</sub>	
Буферная память в мастер-станции или локальной станции	Буферная память со свободным доступом	20 <sub>H</sub>
	Децентрализованные входы	21 <sub>H</sub>
	Децентрализованные выходы	22 <sub>H</sub>
	Децентрализованные регистры	24 <sub>H</sub>
	Специальные маркеры связи	63 <sub>H</sub>
	Специальные регистры связи	64 <sub>H</sub>

- Доступ к операндам в центральном процессоре (код доступа 05<sub>H</sub>)  
Доступ к не названным здесь операндам не возможен. При доступе к битовым операндам указанный адрес должен быть либо равен 0, либо кратен 16.

Операнд		Тип операнда		Код операнда
Обозначение	Символ	битовый	словный	
Входы	X	●		00 <sub>H</sub>
Выходы	Y	●		02 <sub>H</sub>
Маркеры	M	●		03 <sub>H</sub>
Фиксируемые маркеры	L	●		83 <sub>H</sub>
Маркеры связи	B	●		23 <sub>H</sub>
Таймеры (контакт)	T	●		09 <sub>H</sub>
Таймеры (катушка)		●		0A <sub>H</sub>
Таймеры (текущее значение)			●	0C <sub>H</sub>
Счетчики (контакт)	C	●		11 <sub>H</sub>
Счетчики (катушка)		●		12 <sub>H</sub>
Счетчики (текущее значение)			●	14 <sub>H</sub>
Регистры данных	D		●	04 <sub>H</sub>
Регистры связи	W		●	24 <sub>H</sub>
Регистры файлов	R		●	84 <sub>H</sub>

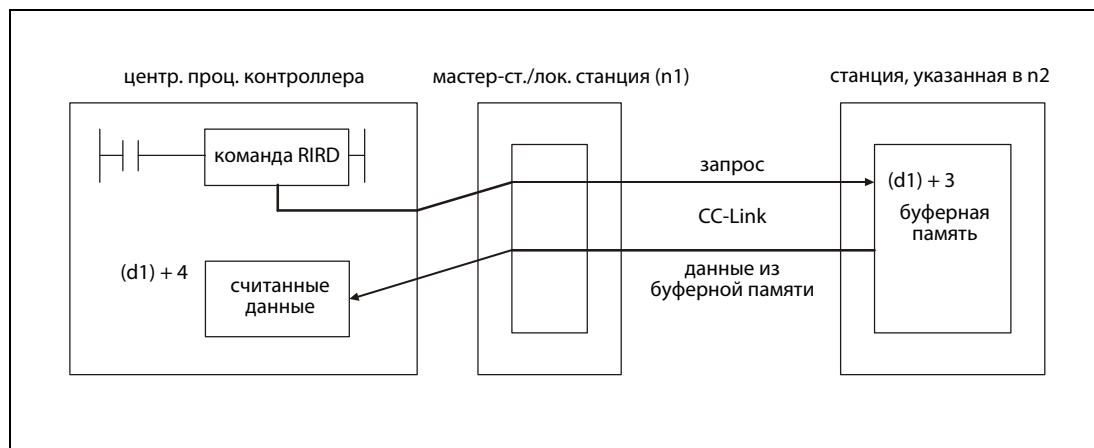
**Принцип действия****Считывание данных из буферной памяти другой станции или из центр. процессора****RIRD Считывание данных**

С помощью команды RIRD можно считать данные из буферной памяти другой станции, подключенной к CC-Link. Если мастер-модуль оснащен программным обеспечением, начиная с версии "J", то возможен и доступ к операндам в центральном процессоре контроллера другой станции.

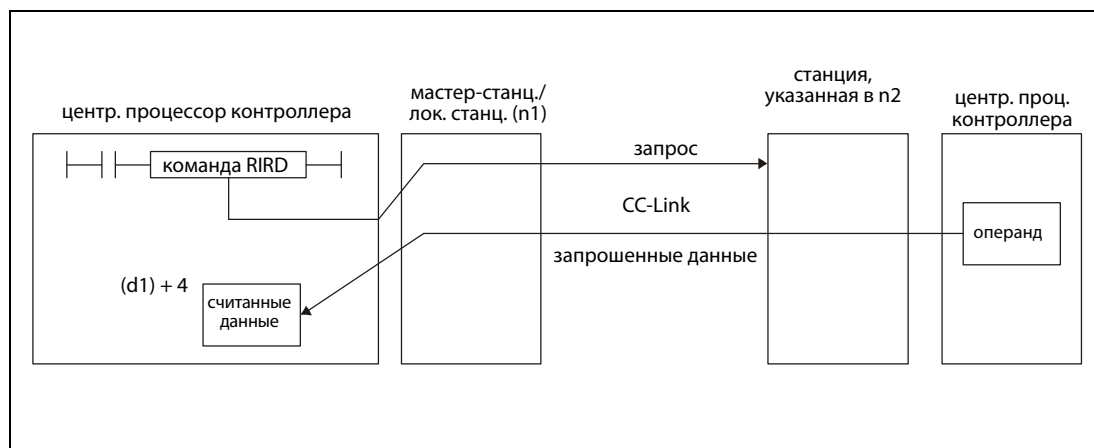
В операнде (d1) + 3 указывается первый адрес в буферной памяти или первый операнд, который требуется считать. Операнд n2 содержит номер другой станции. Эта станция подключена к мастер-станции или локальной станции, указанной в n1. Считанные данные сохраняются в центр. процессоре, обрабатывающем команду RIRD, начиная с операнда (d1) + 4.

В операнде (d1) + 1 указывается, сколько данных требуется передать.

- Функциональная схема в случае версий программного обеспечения от "A" до "H":



- Дополнительная функция, начиная с версии "J" программного обеспечения:

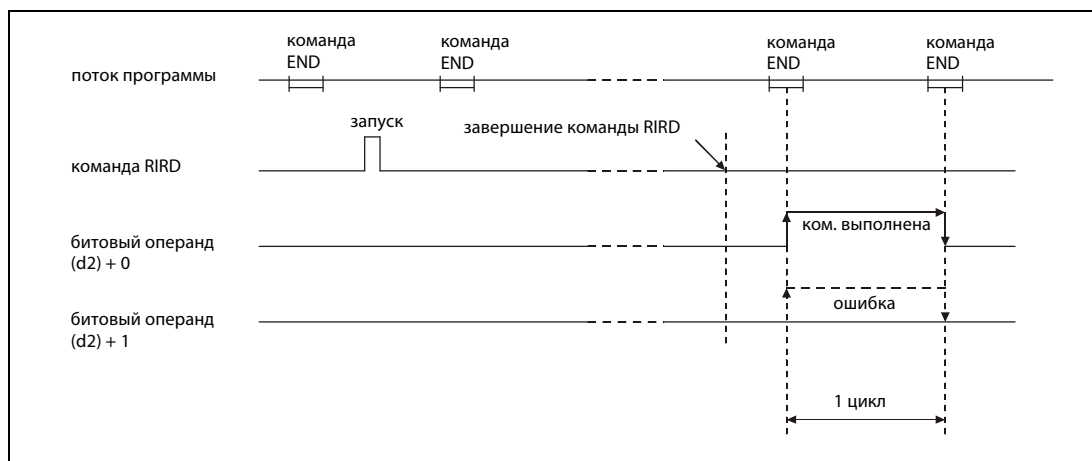


Операнды в (d2) + 0 и (d2) + 1 показывают, выполнена ли команда и возникла ли при ее выполнении ошибка.

- Битовый операнд (d2) + 0 устанавливается при выполнении команды END того цикла, в котором команда RIRD была завершена. При обработке следующей команды END этот битовый операнд снова сбрасывается.
- Битовый операнд (d2) + 1 показывает ошибку при выполнении команды RIRD. При безошибочном выполнении этот битовый операнд остается сброшенным. Если имеется ошибка, то операнд (d2) + 1 устанавливается при выполнении команды END того цикла, в котором команда RIRD была завершена. При обработке следующей команды END этот операнд снова сбрасывается.



На рисунке ниже показана диаграмма изменения сигналов при выполнении команды RIRD:



Команду RIRD можно выполнять одновременно для нескольких станций. Однако к одной и той же интеллектуальной или локальной станции нельзя одновременно обращаться с помощью нескольких команд RIRD из различных станций.

Перед выполнением команды RIRD параметры сети должны быть установлены с помощью команды RLPA.

Если в  $(d1) + 1$  (количество считываемых данных) указан 0 или число вне диапазона 1...480, то при завершении команды RIRD в операнд  $(d2) + 1$  записывается код ошибки.

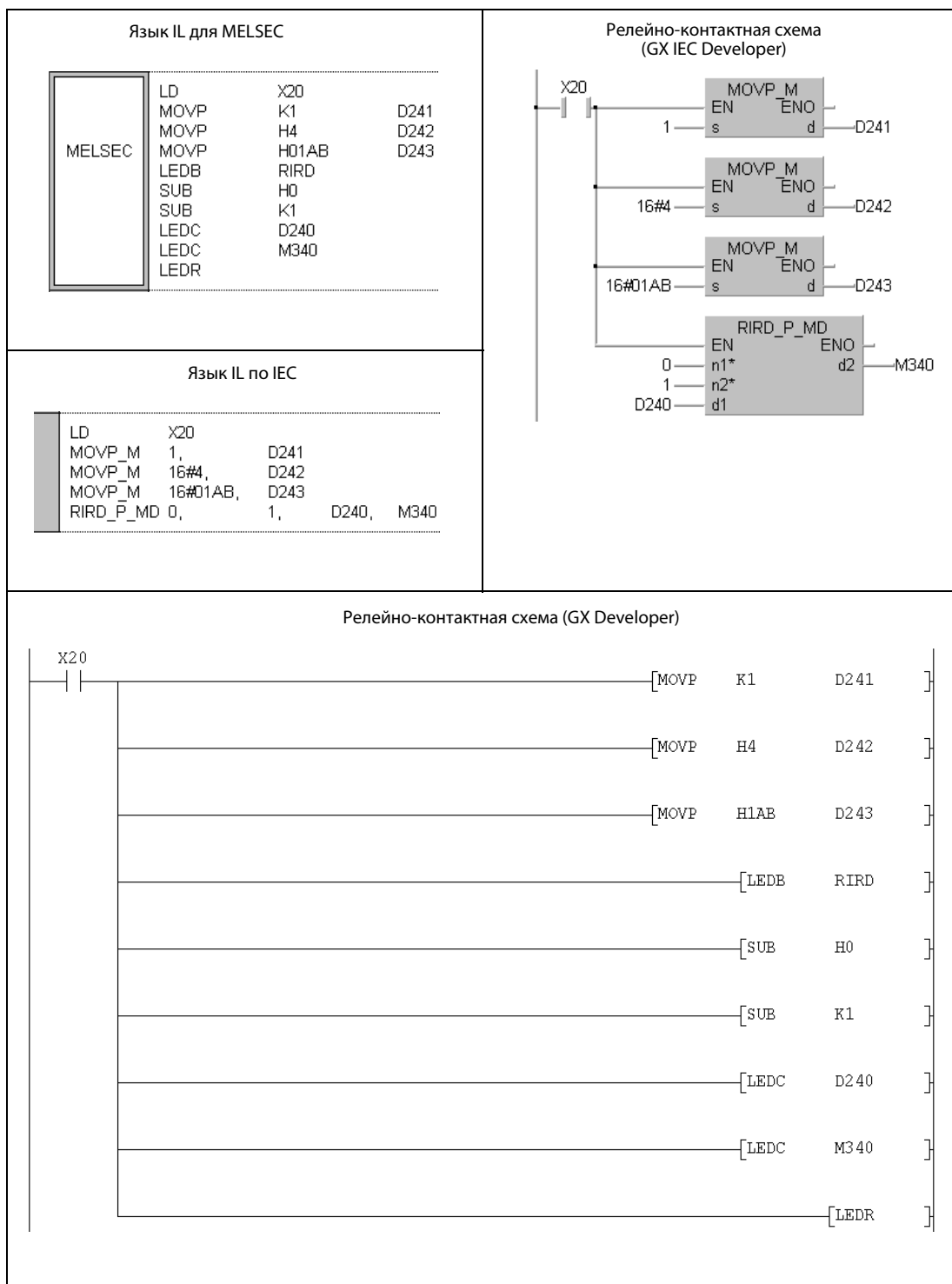
#### Условия выполнения

Если команда RIRD запрограммирована в сочетании с командой LEDA, то команда RIRD выполняется до тех пор, пока условие выполнения команды LEDA включено. При применении команды LEDB процесс считывания выполняется только при положительном фронте условия выполнения.

Учитывайте, что для обработки команды RIRD нужны несколько циклов. Поэтому запускайте очередной процесс считывания лишь после того, как операнд  $(d2) + 0$  показал завершение обработки команда RIRD.

**Пример** RIRD

Следующая программа обрабатывается в центральном процессоре контроллера мастер-станции и считывает из интеллектуальной станции с номером 1 содержимое адреса 1A8<sub>H</sub> буферной памяти. Модуль CC-Link мастер-станции занимает диапазон адресов от X/Y000 до X/Y01F.



Указания по программированию специализированных команд в редакторах MELSEC для контроллеров серии "A" имеются в разделе 3.3 этого руководства.

### 11.5.5 RIRD (серии QnA и "System Q")

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

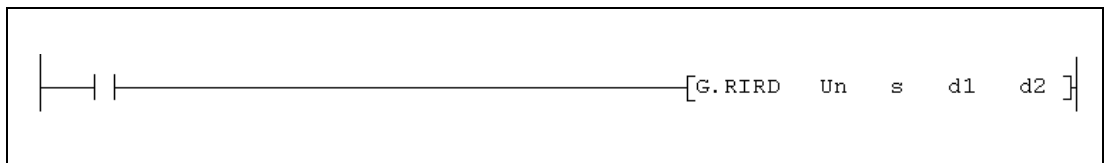
Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□\□□		Спец. модули U□\G□□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	—	SM0	8
d1	—	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		

GX IEC  
Developer



GX  
Developer



## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных
Un	Головной адрес модуля CC-Link на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде 10 <sub>H</sub> .)	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит
Первый операнд области, содержащей информацию для выполнения команды.				
Операнд	Значение	Описание	Диапазон	Устанавливает
(s) + 0	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды: 0000 <sub>H</sub> : безошибочная обработка Любое иное значение кроме 0000 <sub>H</sub> : при обработке возникла ошибка (см. руководство по работе с модулем CC-Link)	—	система
(s) + 1	Номер станции	Номер локальной или интеллектуальной станции, из которой требуется считать данные	от 0 до 64	пользователь
(s) + 2	Код доступа	<ul style="list-style-type: none"> <li>В случае мастер-модуля серии "A" или "Q" с версией программного обеспечения от "A" до "H"</li> </ul> При указании 0004 <sub>H</sub> считывание происходит из буферной памяти интеллектуальной станции. Если требуется считывать данные из буферной памяти локальной станции, введите значение 2004 <sub>H</sub> .	0004 <sub>H</sub> или 2004 <sub>H</sub>	
		<ul style="list-style-type: none"> <li>В случае мастер-модуля серии "A" или "Q" с версией программного обеспечения, начиная с "J", или модуля серии "System Q"</li> </ul> В старший байт этого операнда записывается код операнда. Младший байт операнда служит для выбора цели доступа. Это может быть буферная память модуля CC-Link <sub>H</sub> или центральный процессор (05 <sub>H</sub> ).	старший байт: см. следующую таблицу  младший байт: 04 <sub>H</sub> или 05 <sub>H</sub>	
(s) + 3	Начальный адрес	<ul style="list-style-type: none"> <li>В случае мастер-модуля серии "A" или "Q" с версией программного обеспечения от "A" до "H"</li> </ul> Начальный адрес в буферной памяти	В завис. от станции, к которой осуществляется доступ	
		<ul style="list-style-type: none"> <li>В случае мастер-модуля серии "A" или "Q" с версией программного обеспечения, начиная с "J", или модуля серии "System Q"</li> </ul> Начальный адрес в буф. памяти или первый адрес операнда.		
(s) + 4	Длина данных	Укажите здесь, сколько данных (слов) требуется считать. Длина данных зависит от типа центрального процессора станции, из которой происходит считывание: Серии AnU, QnA, System Q: макс. 480 слов Все прочие центр. процессоры: макс. 32 слова	от 1 до 480 от 1 до 32	

## Переменные

Операнд	Значение		Диапазон	Устанавливает	Тип данных	
d1	Первый адрес области операндов, в которой сохраняются считанные данные.			пользователь	BIN, 16 бит	
d2	Битовый операнд, устанавливаемый на один цикл после выполнения команды RIRD. По адресу (d2)+1 выводится ошибка при выполнении.				бит	
	Операнд	Значение	Описание	Диапазон		Устанавливает
	(d2) + 0	Команда выполнена	Показывает завершение команды RIRD. ВКЛ.: команда выполнена ВЫКЛ.: команда не выполнена	—		система
(d2) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды RIRD. ВКЛ.: команда выполнена с ошибкой ВЫКЛ.: команда выполнена без ошибок	—			

Начиная с версии "J" программного обеспечения главных модулей серий "A" и "Q", а также в модулях "System Q" используются два кода (s) + 2, с помощью которых устанавливается, из какой части буферной памяти происходит считывание или какие операнды центрального процессора определяются:

- Доступ к буферной памяти модуля CC-Link (код доступа 04<sub>H</sub>)

Цель доступа	Код операнда	
Буферная память в интеллектуальных станциях	00 <sub>H</sub>	
Буферная память в мастер-станции или локальной станции	Буферная память со свободным доступом	20 <sub>H</sub>
	Децентрализованные входы	21 <sub>H</sub>
	Децентрализованные выходы	22 <sub>H</sub>
	Децентрализованные регистры	24 <sub>H</sub>
	Специальные маркеры связи	63 <sub>H</sub>
	Специальные регистры связи	64 <sub>H</sub>

- Доступ к операндам в центральном процессоре (код доступа 05<sub>H</sub>)

Доступ к не названным здесь операндам не возможен. При доступе к битовым операндам указанный адрес должен быть либо равен 0, либо кратен 16.

Операнд		Тип операнда		Адресация	Код операнда
Обозначение	Символ	битовый	словный		
Входы	X	●		шестнадцатеричная	00 <sub>H</sub>
Выходы	Y	●			02 <sub>H</sub>
Маркеры	M	●		десятичная	03 <sub>H</sub>
Фиксируемые маркеры	L	●			83 <sub>H</sub>
Маркеры связи	B	●		шестн.	23 <sub>H</sub>
Таймеры (контакт)	T	●		десятичная	09 <sub>H</sub>
Таймеры (катушка)		●			0A <sub>H</sub>
Таймеры (текущее значение)			●		0C <sub>H</sub>
Фиксируемые таймеры (контакт)	ST	●			89 <sub>H</sub>
Фиксируемые таймеры (катушка)		●			8A <sub>H</sub>
Фиксируемые таймеры (текущее значение)			●		8C <sub>H</sub>

Операнд		Тип операнда		Адресация	Код операнда
Обозначение	Символ	битовый	словный		
Счетчики (контакт)	C	●		десятичная	11 <sub>H</sub>
Счетчики (катушка)		●			12 <sub>H</sub>
Счетчики (текущее значение)			●		14 <sub>H</sub>
Регистры данных	D		●		04 <sub>H</sub>
Регистры связи	W		●	шестн.	24 <sub>H</sub>
Регистры файлов	R		●	десятичная	84 <sub>H</sub>
Специальные маркеры связи	SB	●		шестнадцатеричная	63 <sub>H</sub>
Специальные регистры связи	SW		●		64 <sub>H</sub>
Специальные маркеры	SM	●		десятичная	43 <sub>H</sub>
Специальные регистры	SD		●		44 <sub>H</sub>

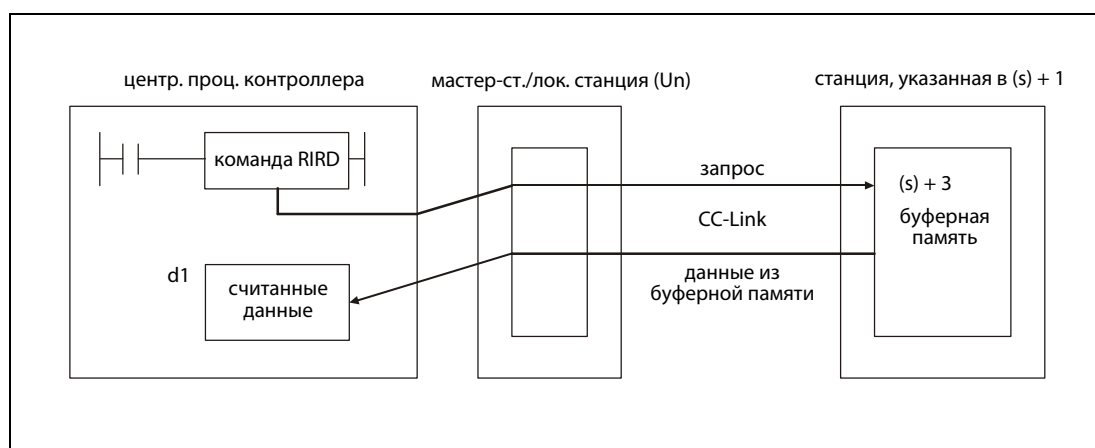
**Принцип действия****Считывание данных из буферной памяти другой станции или из центр. процессора****RIRD Считывание данных**

С помощью команды RIRD можно считать данные из буферной памяти другой станции, подключенной к CC-Link. Из мастер-модуля CC-Link с версией программного обеспечения "J" или выше, а также из модулей CC-Link серии "System Q" возможен также доступ к операндам центрального процессора контроллера в другой станции CC-Link.

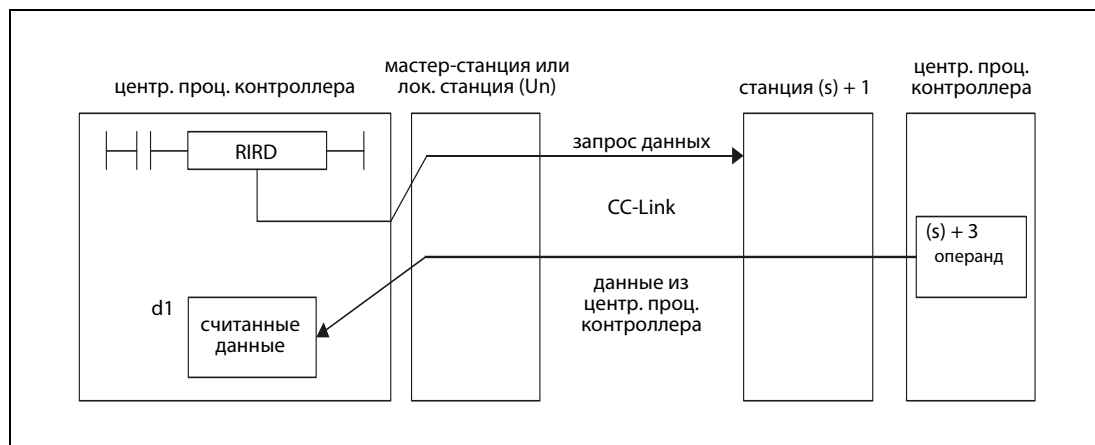
В операнде (s) + 3 указывается первый адрес буферной памяти или первый операнд, который требуется считать. Операнд (s) + 1 содержит номер другой станции. Эта станция подключена к мастер-станции, указанной в Un. Считанные данные сохраняются в центральном процессоре, обрабатывающем команду RIRD, начиная с операнда, указанного в d1.

В операнде (s) + 4 указано количество передаваемых данных.

- Функциональная схема при считывании из буферной памяти модуля CC-Link



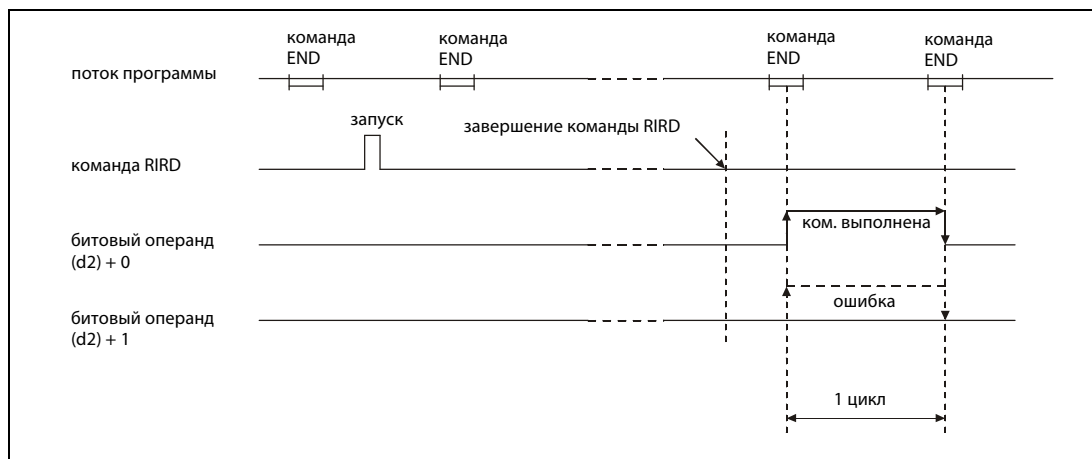
- Функциональная схема при доступе к операндам центрального процессора контроллера другой станции CC-Link



Операнды в (d2) + 0 и (d2) + 1 показывают, выполнена ли команда и возникла ли при ее выполнении ошибка.

- Битовый операнд (d2) + 0 устанавливается при выполнении команды END того цикла, в котором команда RIRD была завершена. При обработке следующей команды END этот битовый операнд снова сбрасывается.
- Битовый операнд (d2) + 1 показывает ошибку при выполнении команды RIRD. При безошибочном выполнении этот битовый операнд остается сброшенным. Если имеется ошибка, то операнд (d2) + 1 устанавливается при выполнении команды END того цикла, в котором команда RIRD была завершена. При обработке следующей команды END этот операнд снова сбрасывается.

На рисунке ниже показана диаграмма изменения сигналов при выполнении команды RIRD:



Команду RIRD можно выполнять одновременно для нескольких станций. Однако к интеллектуальной или локальной станции невозможно одновременно обращаться с помощью команд RIRD из различных других станций.

#### Источники ошибок

Если при выполнении команды RIRD возникла ошибка, то устанавливается флаг ошибки SMO и в специальный регистр SD0 записывается код ошибки.

- Указанный в Un модуль не является специальным модулем.  
(код ошибок в процессорах QnA: 2110, в процессорах "System Q": 2112)
- Выполняемая команда не поддерживается (код ошибки: 4002).
- В команде указано недействительное количество операндов (код ошибки: 4003).
- В команде указан недопустимый операнд (код ошибки: 4004).
- Область операндов, определенная с помощью s, содержит недопустимые данные (код ошибки: 4100).
- Количество данных превышает допустимый диапазон (код ошибки: 4101).
- Сохраненные данные или константы, переданные с помощью этой команды, превышают допустимый диапазон (код ошибки: 4101).
- В случае серии QnA: используется слишком много специализированных команд для CC-Link (код ошибки: 4107).
- В случае серии QnA: не настроены параметры связи (код ошибки: 4108).

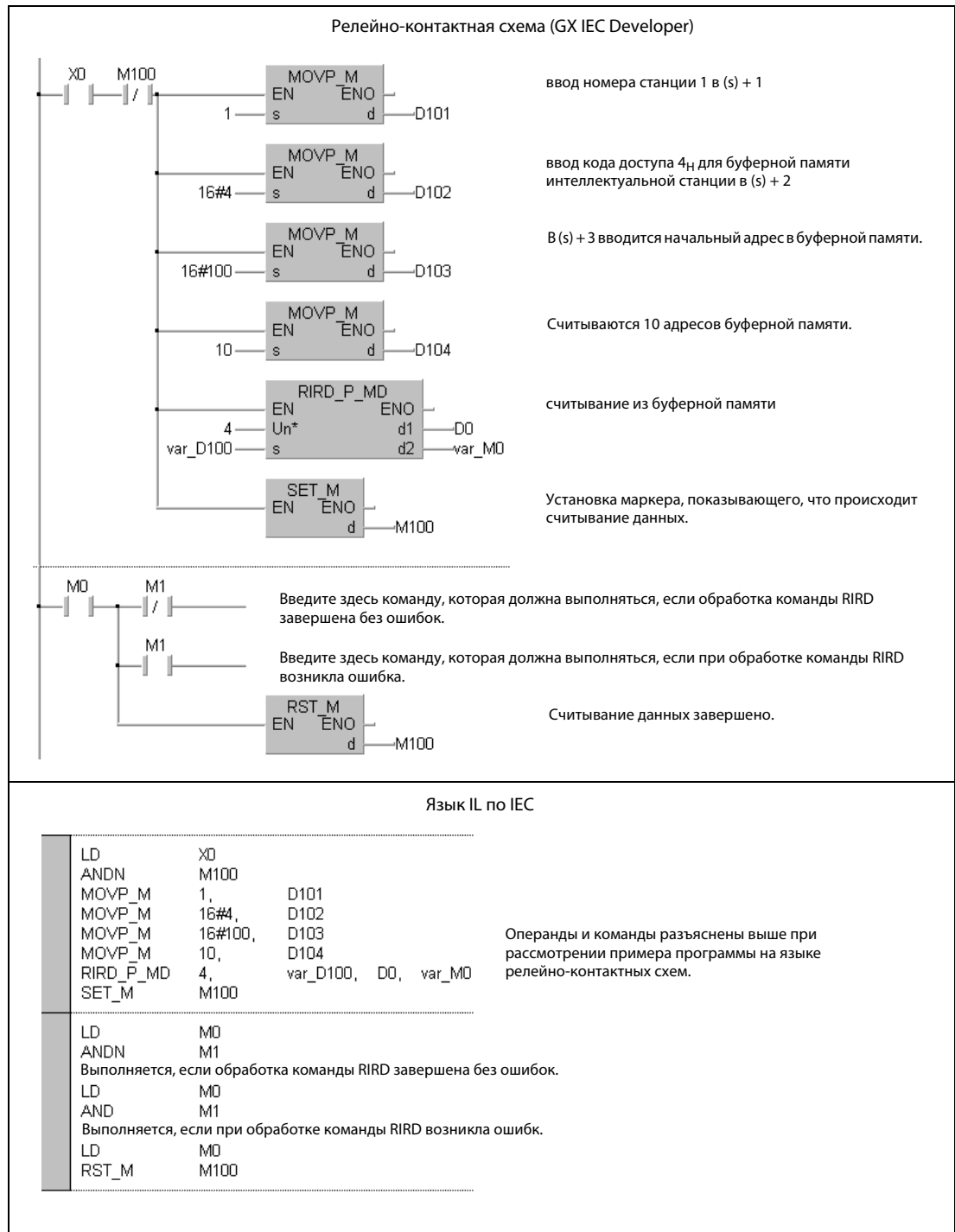


**Пример**

**RIRD**

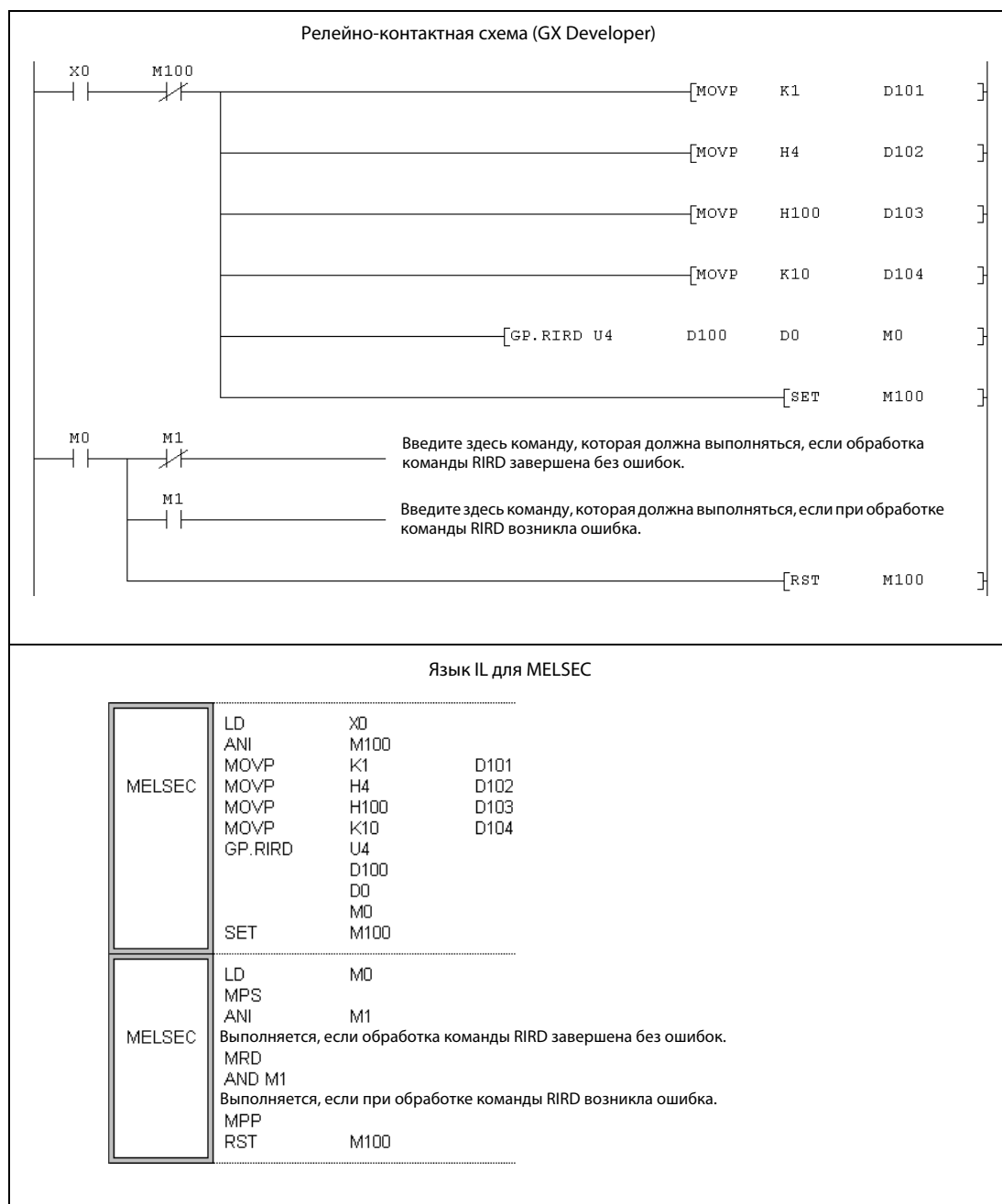
Следующая программа обрабатывается в контроллере мастер-станции. Если вход X0 установлен, из интеллектуальной станции с номером 1 считывается содержимое 10 адресов буферной памяти. Процесс считывания начинается с адреса 100<sub>H</sub> буферной памяти. Считанные данные сохраняются, начиная с регистра D0. Модуль CC-Link мастер-станции имеет головной адрес X/Y40.

- Редакторы по стандарту IEC (на следующей странице этот пример воспроизведен на языке списка инструкций MELSEC и языке LD в среде GX Developer)



**ПРИМЕЧАНИЕ** В редакторах по стандарту IEC среды GX IEC Developer переменные должны быть объявлены в заголовке программного компонента (POU). Если переменные не были объявлены, то при проверке или компилировании программы возникает сообщение об ошибке. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

- Список инструкций MELSEC и релейно-контактная схема в среде GX Developer  
Операнды и команды разъяснены на предыдущей странице при рассмотрении примера программы на языке LD в среде GX IEC Developer.



### 11.5.6 RIWT (серия "A")

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

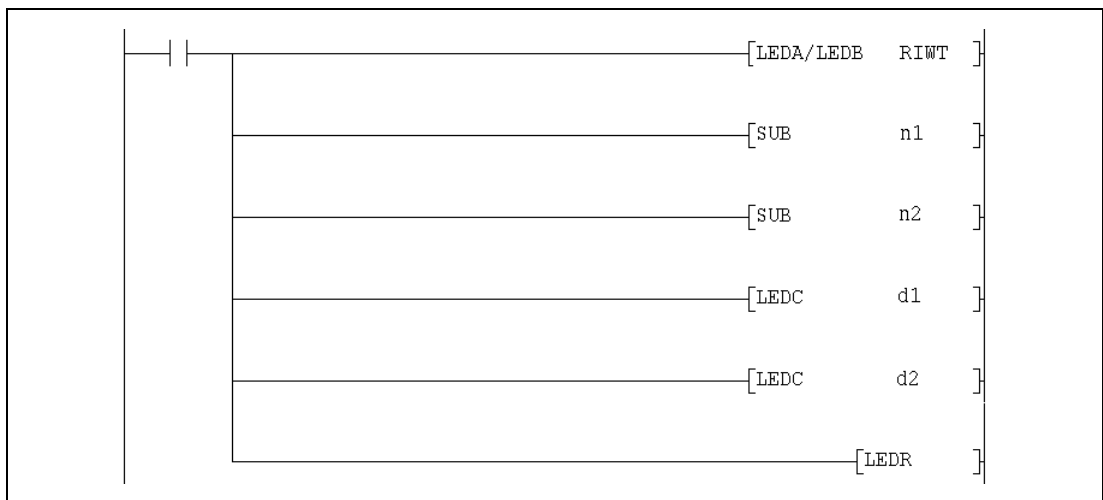
Операнды MELSEC A

	Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9011			
	битовые						словные (16 бит)						Константы		Указатели							Уровень		
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V							K	H (16#)
n1																	●	●				26		
n2																	●	●						
d1							●	●	●	●	●													●
d2	●	●	●	●	●																			

GX IEC Developer



GX Developer



Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных
n1	Головной адрес модуля CC-Link на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде 10H.)	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит
n2	Номер станции, в которую передаются данные Диапазон: если команда RIWT выполняется в мастер-станции: от 1 до 64; если команда RIWT выполняется в локальной станции: от 0 до 64	см. разъяснение рядом	пользователь	BIN, 16 бит

## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных		
d1	Первый операнд области, содержащей информацию для выполнения команды и передаваемые данные				BIN, 16 бит	
	Операнд	Значение	Описание	Диапазон		Устанавливает
	(d1) + 0	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды: 0000 <sub>H</sub> : безошибочная обработка Любое иное значение кроме 0000 <sub>H</sub> : при обработке возникла ошибка. Более подробные указания, относящиеся к кодам ошибок, имеются в руководстве по эксплуатации модуля CC-Link	—		система
	(d1) + 1	Длина данных	Укажите здесь, сколько данных (слов) требуется передать. Длина данных зависит от типа центрального процессора станции, в которую передаются данные: Серии AnU, QnA, System Q: макс. 480 слов Все прочие центр. процессоры: макс. 10 слов	от 1 до 480 от 1 до 10		пользователь
	(d1) + 2	Код доступа	<ul style="list-style-type: none"> <li>В случае мастер-модуля с версией программного обеспечения от "А" до "Н"</li> </ul> При указании 0004 <sub>H</sub> запись осуществляется в буферную память интеллектуальной станции. При указании 2004 <sub>H</sub> запись осуществляется в буферную память локальной станции.	0004 <sub>H</sub> или 2004 <sub>H</sub>		
			<ul style="list-style-type: none"> <li>В случае мастер-модуля с версией программного обеспечения, начиная с "J"</li> </ul> В старший байт этого операнда записывается код операнда. Младший байт операнда служит для выбора цели доступа. Это может быть буферная память модуля CC-Link <sub>H</sub> или центральный процессор (05 <sub>H</sub> ).	старший байт: см. следующую таблицу  младший байт: 04 <sub>H</sub> или 05 <sub>H</sub>		
	(d1) + 3	Начальный адрес	<ul style="list-style-type: none"> <li>В случае мастер-модуля с версией программного обеспечения от "А" до "Н"</li> </ul> Начальный адрес в буферной памяти	В зависимости от станции, к которой осуществляется доступ.		
			<ul style="list-style-type: none"> <li>В случае мастер-модуля с версией программного обеспечения, начиная с "J"</li> </ul> Начальный адрес в буферной памяти или первый адрес операнда.			
от (d1) + 4 до (d1) + n	Область памяти для передаваемых данных.	Укажите размер этой области в (d1) + 1.	—	пользователь		

## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных
d2	Битовый операнд, устанавливаемый на один цикл после выполнения команды RIWT. По адресу (d2) + 1 выводится ошибка при выполнении.			
	Операнд	Значение	Описание	Диапазон
	(d2) + 0	Команда выполнена	Показывает завершение команды. ВКЛ.: команда выполнена ВыКЛ.: команда не выполнена	—
(d2) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды. ВКЛ.: команда выполнена с ошибкой ВыКЛ.: команда выполнена без ошибок	—	система
				бит

Начиная с версии "J" программного обеспечения главного модуля, для указания целевой области (буферная память или операнды центрального процессора) используются два кода, содержащиеся в операнде (d1) + 2:

- Доступ к буферной памяти модуля CC-Link (код доступа 04<sub>H</sub>)

Цель доступа	Код операнда	
Буферная память в интеллектуальных станциях	00 <sub>H</sub>	
Буферная память в мастер-станции или локальной станции	Буферная память со свободным доступом	20 <sub>H</sub>
	Децентрализованные входы	21 <sub>H</sub>
	Децентрализованные выходы	22 <sub>H</sub>
	Децентрализованные регистры	24 <sub>H</sub>
	Специальные маркеры связи	63 <sub>H</sub>
	Специальные регистры связи	64 <sub>H</sub>

- Доступ к операндам в центральном процессоре (код доступа 05<sub>H</sub>)

Доступ к не названным здесь операндам не возможен. При доступе к битовым операндам указанный адрес должен быть либо равен 0, либо кратен 16.

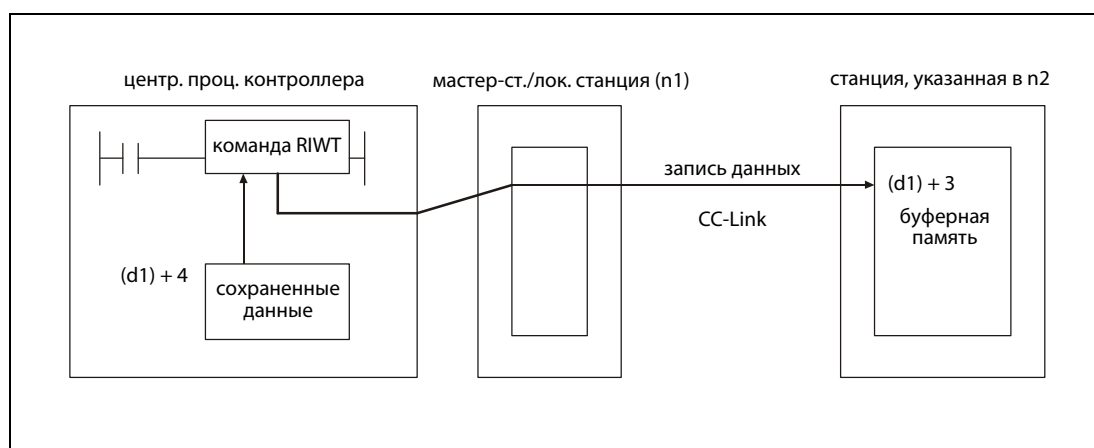
Операнд	Тип операнда		Код операнда	
	Обозначение	Символ		битовый
Входы	X	●		01 <sub>H</sub>
Выходы	Y	●		02 <sub>H</sub>
Маркеры	M	●		03 <sub>H</sub>
Фиксируемые маркеры	L	●		83 <sub>H</sub>
Маркеры связи	B	●		23 <sub>H</sub>
Таймеры (контакт)	T	●		09 <sub>H</sub>
Таймеры (катушка)		●		0A <sub>H</sub>
Таймеры (текущее значение)			●	0C <sub>H</sub>
Счетчики (контакт)	C	●		11 <sub>H</sub>
Счетчики (катушка)		●		12 <sub>H</sub>
Счетчики (текущее значение)			●	14 <sub>H</sub>
Регистры данных	D		●	04 <sub>H</sub>
Регистры связи	W		●	24 <sub>H</sub>
Регистры файлов	R		●	84 <sub>H</sub>

**Принцип действия****Запись данных в буферную память другой станции или в другой центральный процессор****RIWT      Запись данных**

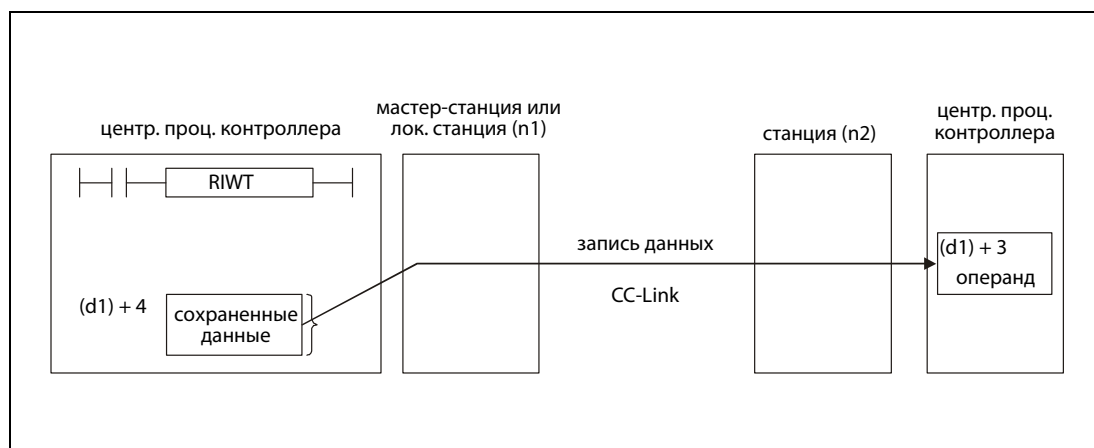
С помощью команды RIWT данные можно передать в буферную память другой станции, подключенной к CC-Link. Если мастер-модуль оснащен программным обеспечением, начиная с версии "J", то возможен и доступ к операндам в центральном процессоре контроллера другой станции.

Операнд  $n2$  содержит номер другой станции. Эта станция подключена к главной/локальной станции, которая указана в  $n1$ . Передаваемые данные предварительно сохранены в центральном процессоре, обрабатывающем команду RIWT, начиная с операнда  $(d1) + 4$ . В операнде  $(d1) + 1$  указывается, сколько данных требуется передать. В операнде  $(d1) + 3$  указывается первый адрес в буферной памяти или первый операнд, который требуется перезаписать.

- Функциональная схема для версий программного обеспечения от "A" до "H":



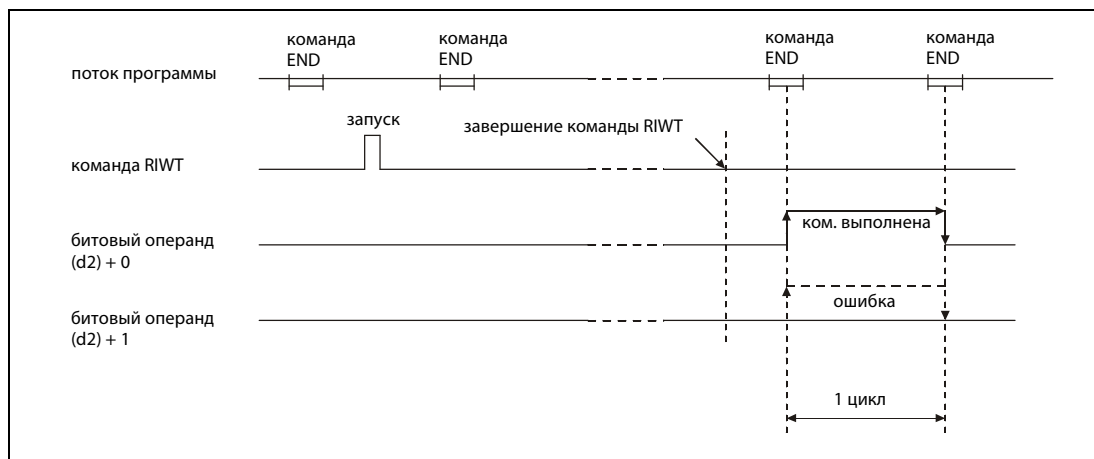
- Дополнительная функция, начиная с версии "J" программного обеспечения:



Операнды в  $(d2) + 0$  и  $(d2) + 1$  показывают, выполнена ли команда и возникла ли при ее выполнении ошибка.

- Битовый операнд  $(d2) + 0$  устанавливается при выполнении команды END того цикла, в котором команда RIWT была завершена. При обработке следующей команды END этот битовый операнд снова сбрасывается.
- Битовый операнд  $(d2) + 1$  показывает ошибку при выполнении команды RIWT. При безошибочном выполнении этот битовый операнд остается сброшенным. Если имеется ошибка, то операнд  $(d2) + 1$  устанавливается при выполнении команды END того цикла, в котором команда RIWT была завершена. При обработке следующей команды END этот операнд снова сбрасывается.

На рисунке ниже показана диаграмма изменения сигналов при выполнении команды RIWT:



Команду RIWT можно выполнять одновременно для нескольких станций. Однако к одной и той же интеллектуальной или локальной станции невозможно обращаться одновременно из различных станций с помощью нескольких команд RIWT.

Перед выполнением команды RIWT должны быть установлены параметры сети с помощью команды RLPA.

Если количество передаваемых данных, указанное в  $(d1) + 1$ , равно 0 или находится вне диапазона от 1 до 480, то по окончании выполнения команды RIWT в операнд  $(d2) + 1$  записывается сообщение об ошибке.

#### Условия выполнения

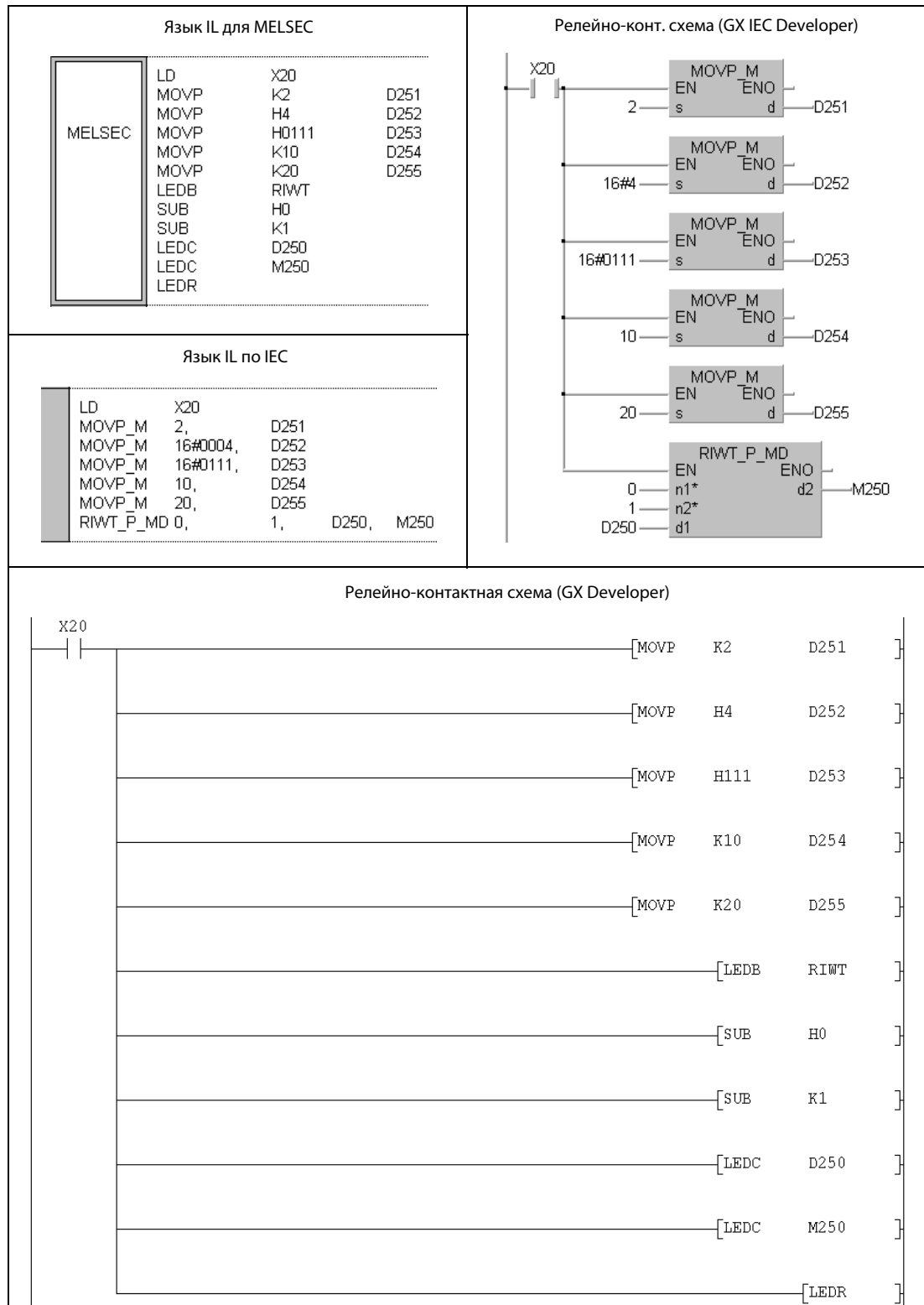
Если команда RIWT запрограммирована в сочетании с командой LEDA, то команда RIWT выполняется до тех пор, пока условие выполнения команды LEDA включено. При применении команды LEDB передача данных выполняется только при отрицательном фронте условия выполнения.

Учитывайте, что для обработки команды RIWT нужны несколько циклов. Поэтому запускайте очередную передачу данных лишь в после того, как операнд  $(d2) + 0$  показал завершение обработки команды RIWT.

**Пример**

**RIWT**

Следующая программа обрабатывается в контроллере мастер-станции и записывает значение 10 в адрес 111<sub>H</sub> буферной памяти и значение 20 в адрес 112<sub>H</sub> буферной памяти интеллектуальной станции с номером 1. Модуль CC-Link мастер-станции занимает диапазон адресов от X/Y000 до X/Y01F.



Указания по программированию специализированных команд в редакторах MELSEC для контроллеров серии "A" имеются в разделе 3.3 этого руководства.



### 11.5.7 RIWT (серии QnA и "System Q")

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

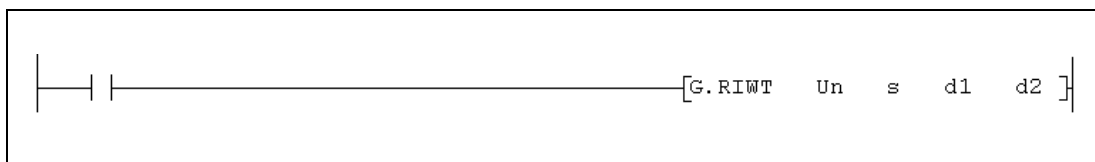
Операнды  
MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s	—	●	●	—	—	—	—	—	—	SM0	8
d1	—	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		

GX IEC  
Developer



GX  
Developer



Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных
Un	Головной адрес модуля CC-Link на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде 10 <sub>н</sub> .)	от 0 до FE <sub>н</sub>	пользователь	BIN, 16 бит
Первый операнд области, содержащей информацию для выполнения команды.				
Операнд	Значение	Описание	Диапазон	Устанавливает
(s) + 0	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды: 0000 <sub>н</sub> : безошибочная обработка Любое иное значение кроме 0000 <sub>н</sub> : при обработке возникла ошибка (см. руководство по пользованию модулем CC-Link)	—	система
(s) + 1	Номер станции	Номер локальной или интеллектуальной станции, в которую требуется передать данные.	от 0 до 64	пользователь
(s) + 2	Код доступа	<ul style="list-style-type: none"> <li>В случае мастер-модуля серии "А" или "Q" с версией программного обеспечения от "А" до "Н"</li> </ul> При указании 0004 <sub>н</sub> запись осуществляется в буферную память интеллектуальной станции. При указании 2004 <sub>н</sub> запись осуществляется в буферную память локальной станции.	0004 <sub>н</sub> или 2004 <sub>н</sub>	
		<ul style="list-style-type: none"> <li>В случае мастер-модуля серии "А" или "Q" с версией программного обеспечения, начиная с "J", или модуля серии "System Q"</li> </ul> В старший байт этого операнда записывается код операнда. Младший байт операнда служит для выбора цели доступа. Это может быть буферная память модуля CC-Link <sub>н</sub> или центральный процессор (05 <sub>н</sub> ).	старший байт: см. следующую таблицу  младший байт: 04 <sub>н</sub> или 05 <sub>н</sub>	
(s) + 3	Начальный адрес	<ul style="list-style-type: none"> <li>В случае мастер-модуля серии "А" или "Q" с версией программного обеспечения от "А" до "Н"</li> </ul> Начальный адрес в буферной памяти	В завис. от станции, к которой осуществляется доступ.	
		<ul style="list-style-type: none"> <li>В случае мастер-модуля серии "А" или "Q" с версией программного обеспечения, начиная с "J", или модуля серии "System Q"</li> </ul> Начальный адрес в буферной памяти или первый адрес операнда.		
(s) + 4	Количество записываемых данных	Укажите здесь, сколько данных (слов) требуется передать. Длина данных зависит от типа центрального процессора станции, в которую передаются данные: Серии AnU, QnA, System Q: макс. 480 слов Все прочие центр. процессоры: макс. 10 слов	от 1 до 480 от 1 до 10	BIN, 16 бит

## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных	
d1	Первый адрес области операндов, в котором сохранены передаваемые данные.		пользователь	BIN, 16 бит	
d2	Битовый операнд, устанавливаемый на один цикл после выполнения команды RIWT. По адресу (d2) + 1 выводится ошибка при выполнении.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(d2) + 0	Команда выполнена	Показывает завершение команды RIWT. ВКЛ.: команда выполнена Выкл.: команда не выполнена	—	система
(d2) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды RIWT. ВКЛ.: команда выполнена с ошибкой Выкл.: команда выполнена без ошибок	—		

Начиная с версии "J" программного обеспечения главного модуля серий A/Q, а также в модулях серии "System Q" для указания целевой области записи (буферная память или операнды центрального процессора) имеются два кода, указываемые в операнде (s) + 2:

- Доступ к буферной памяти модуля CC-Link (код доступа 04<sub>H</sub>)

Цель доступа	Код операнда	
Буферная память в интеллектуальных станциях	00 <sub>H</sub>	
Буферная память в мастер-станции или локальной станции	Буферная память со свободным доступом	20 <sub>H</sub>
	Децентрализованные входы	21 <sub>H</sub>
	Децентрализованные выходы	22 <sub>H</sub>
	Децентрализованные регистры	24 <sub>H</sub>
	Специальные маркеры связи	63 <sub>H</sub>
	Специальные регистры связи	64 <sub>H</sub>

- Доступ к операндам в центральном процессоре (код доступа 05<sub>H</sub>)  
Доступ к не названным здесь операндам не возможен. При доступе к битовым операндам указанный адрес должен быть либо равен 0, либо кратен 16.

Операнд		Тип операнда		Адресация	Код операнда
Обозначение	Символ	битовый	словный		
Входы	X	●		шестнадцатеричная	01 <sub>H</sub>
Выходы	Y	●			02 <sub>H</sub>
Маркеры	M	●		десятичная	03 <sub>H</sub>
Фиксируемые маркеры	L	●			83 <sub>H</sub>
Маркеры связи	B	●		шестн.	23 <sub>H</sub>
Таймеры (контакт)	T	●		десятичная	09 <sub>H</sub>
Таймеры (катушка)		●			0A <sub>H</sub>
Таймеры (текущее значение)			●		0C <sub>H</sub>
Фиксируемые таймеры (контакт)	ST	●			89 <sub>H</sub>
Фиксируемые таймеры (катушка)		●			8A <sub>H</sub>
Фиксируемые таймеры (текущее значение)			●		8C <sub>H</sub>

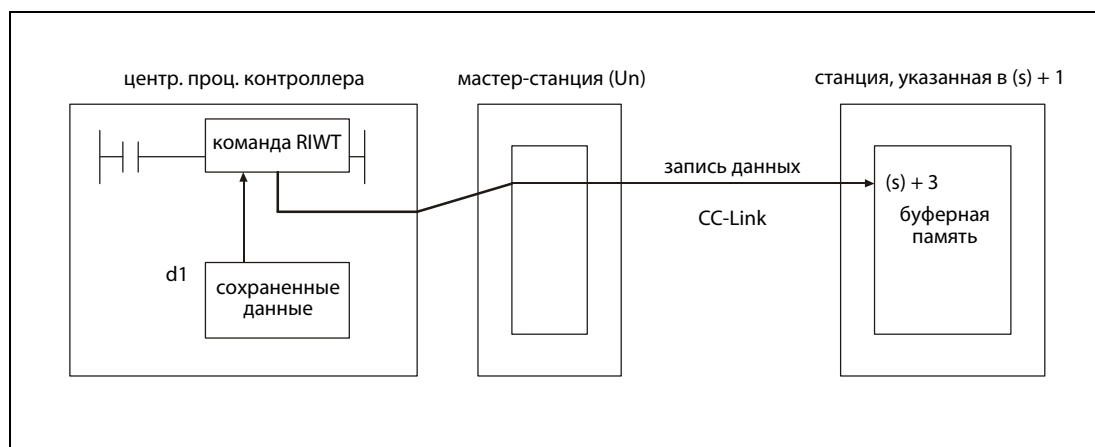
Операнд		Тип операнда		Адресация	Код операнда
Обозначение	Символ	битовый	словный		
Счетчики (контакт)	C	●		десятичная	11 <sub>H</sub>
Счетчики (катушка)		●			12 <sub>H</sub>
Счетчики (текущее значение)			●		14 <sub>H</sub>
Регистры данных	D		●		04 <sub>H</sub>
Регистры связи	W		●	шестн.	24 <sub>H</sub>
Регистры файлов	R		●	десятичная	84 <sub>H</sub>
Специальные маркеры связи	SB	●		шестнадцатеричная	63 <sub>H</sub>
Специальные регистры связи	SW		●		64 <sub>H</sub>
Специальные маркеры	SM	●		десятичная	43 <sub>H</sub>
Специальные регистры	SD		●		44 <sub>H</sub>

**Принцип действия****Запись данных в буферную память другой станции или в другой центральный процессор****RIWT      Запись данных**

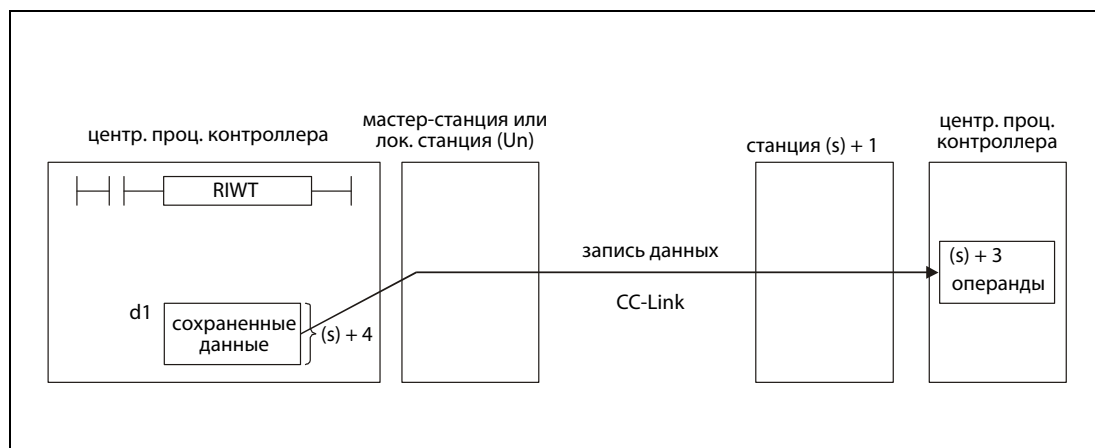
С помощью команды RIWT данные можно передать в буферную память другой станции, подключенной к CC-Link. Из главных модулей CC-Link, оснащенных версией программного обеспечения "J" и выше, а также из модулей CC-Link серии "System Q" возможен также доступ к операндам центрального процессора контроллера другой станции.

Операнд  $(s) + 1$  содержит номер другой станции. Эта станция подключена к мастер-станции, указанной в  $U_n$ . Местонахождение передаваемых данных устанавливается в  $d1$ . В  $(s) + 2$  в закодированном виде указывается, куда записываются данные – в буферную память или в центральный процессор, и какие операнды при этом затрагиваются. Начальный адрес в буферной памяти или первый адрес операнда сохраняется в  $(s) + 3$ . В операнде  $(s) + 4$  указывается, сколько данных требуется передать.

- Функциональная схема при записи в буферную память модуля CC-Link



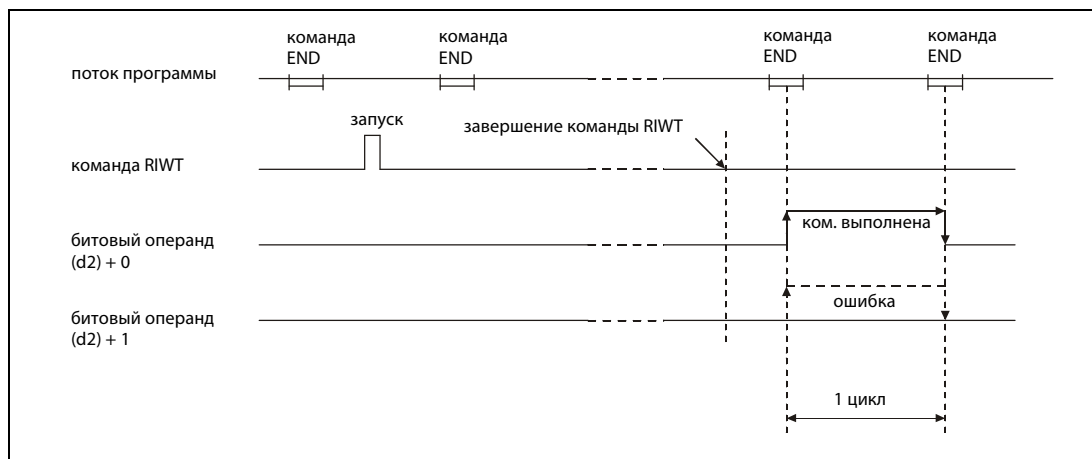
- Функциональная схема при доступе к операндам центрального процессора контроллера другой станции CC-Link



Операнды в  $(d2) + 0$  и  $(d2) + 1$  показывают, выполнена ли команда и возникла ли при ее выполнении ошибка.

- Битовый операнд  $(d2) + 0$  устанавливается при выполнении команды END того цикла, в котором команда RIWT была завершена. При обработке следующей команды END этот битовый операнд снова сбрасывается.
- Битовый операнд  $(d2) + 1$  показывает ошибку при выполнении команды RIWT. При безошибочном выполнении этот битовый операнд остается сброшенным. При наличии ошибки он устанавливается при выполнении команды END того цикла, в котором была завершена команда RIWT. При обработке следующей команды END битовый операнд  $(d2) + 1$  снова сбрасывается.

На рисунке ниже показана диаграмма изменения сигналов при выполнении команды RIWT:



Команду RIWT можно выполнять одновременно для нескольких станций. Однако к интеллектуальной или локальной станции невозможно одновременно обращаться из различных других станций с помощью нескольких команд RIWT.

#### Источники ошибок

Если при выполнении команды RIWT возникла ошибка, то устанавливается флаг ошибки SMO и в специальный регистр SD0 записывается код ошибки.

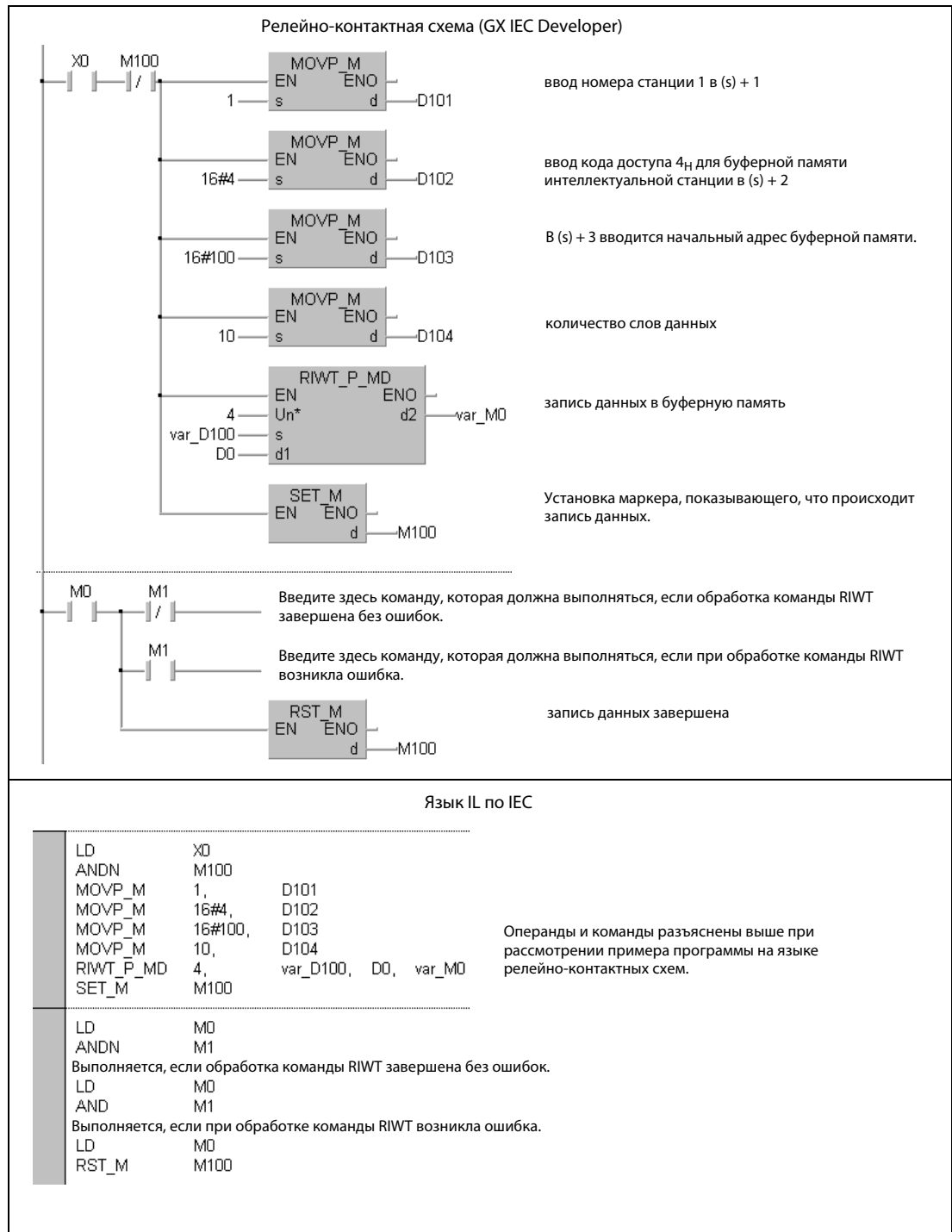
- Указанный в Un модуль не является специальным модулем (код ошибки: 2112).
- Выполняемая команда не поддерживается (код ошибки: 4002).
- В команде указано недействительное количество операндов (код ошибки: 4003).
- В команде указан недопустимый операнд (код ошибки: 4004).
- Область операндов, определенная с помощью s, содержит недопустимые данные (код ошибки: 4100).
- Количество данных превышает допустимый диапазон (код ошибки: 4101).
- Сохраненные данные или константы, переданные с помощью этой команды, превышают допустимый диапазон (код ошибки: 4101).

**Пример**

**RIWT**

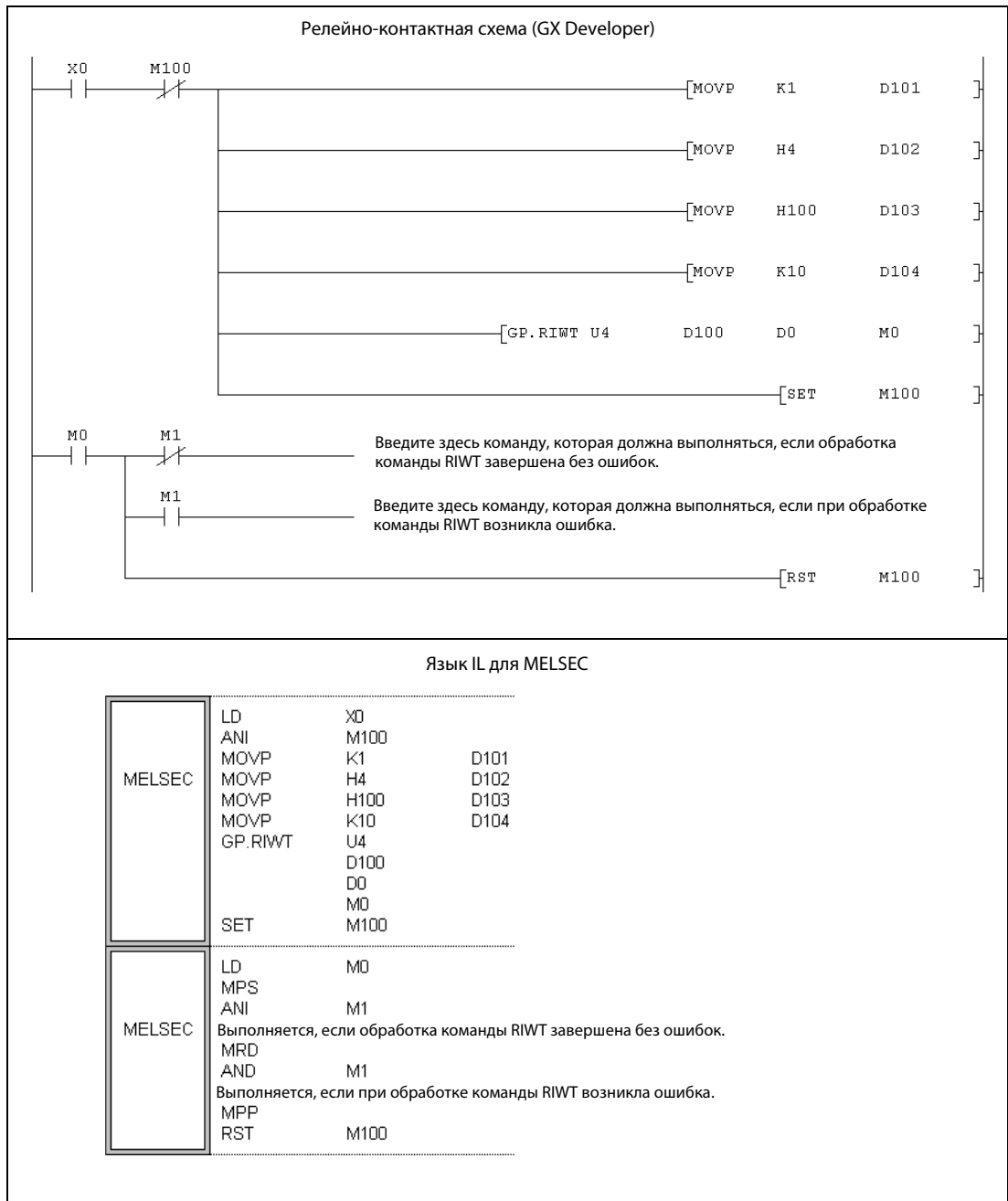
Следующая программа обрабатывается в контроллере мастер-станции. При установке входа X0 в десять адресов буферной памяти интеллектуальной станции № 1 записываются данные, начиная с адреса 100<sub>H</sub>. Передаваемые данные сохранены в центральном процессоре, начиная с регистра D0. Модуль CC-Link мастер-станции имеет головной адрес X/Y40.

- Редакторы по стандарту IEC (на следующей странице этот пример воспроизведен на языке списка инструкций MELSEC и языке LD в среде GX Developer)



**ПРИМЕЧАНИЕ** В редакторах по стандарту IEC среды GX IEC Developer переменные должны быть объявлены в заголовке программного компонента (POU). Если переменные не были объявлены, то при проверке или компилировании программы возникает сообщение об ошибке. Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

- Список инструкций MELSEC и релейно-контактная схема в среде GX Developer  
Операнды и команды разъяснены на предыдущей странице при рассмотрении примера программы на языке LD в среде GX IEC Developer.





### 11.5.8 RIRCV (серия "A")

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

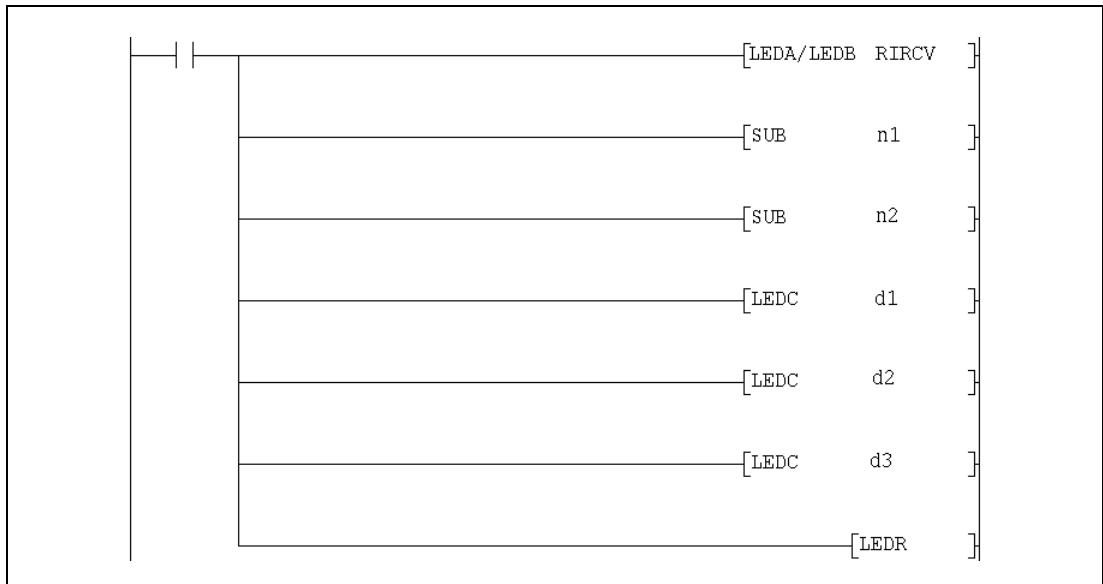
Операнды MELSEC A

	Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9011			
	битовые						словные (16 бит)						Константы		Указатели							Уровень		
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V							K	H (16#)
n1																	●	●				29		
n2																●	●							
d1								●	●	●	●	●												●
d2								●	●	●	●	●												
d3	●	●	●	●	●																			

GX IEC Developer



GX Developer



## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных	
n1	Головной адрес мастер-модуля CC-Link на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде 10 <sub>H</sub> )	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит	
n2	Номер интеллектуальной станции, из которой требуется считать данные.	от 1 до 64	пользователь	BIN, 16 бит	
d1	Первый операнд области, содержащей информацию для выполнения команды и сохранения считанных данных.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(d1) + 0	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды: 0000 <sub>H</sub> : безошибочная обработка Любое иное значение кроме 0000 <sub>H</sub> : при обработке возникла ошибка. Более подробные указания, относящиеся к кодам ошибок, имеются в руководстве по эксплуатации модуля CC-Link	—	система
	(d1) + 1	Длина данных	Укажите здесь, сколько данных (слов) требуется считать. Указанное значение должно находиться в пределах длины буферной памяти интеллектуальной станции и размера принимающего буфера мастер-станции.	от 1 до 480	пользователь
	(d1) + 2	Код доступа	Укажите значение 0004 <sub>H</sub> (считывание из буферной памяти интеллектуальной станции).	0004 <sub>H</sub>	
	(d1) + 3	Выявление ошибок	Выберите операнд, показывающий ошибку при выполнении команды RIRCV: 0: операнд d1 1: операнд RX + 1	0 или 1	
	(d1) + 4	Начальный адрес	Начальный адрес в буферной памяти интеллектуальной станции	В завис. от станции, к которой осуществл. доступ.	
	от (d1) + 5 до (d1) + n	Область памяти для считанных данных	Размер этой области определяется количеством данных, указанных в (d1) + 1.	—	система
d2	Указание операндов связи, используемых для квитирования связи.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(d2) + 0	Децентрализованный вход (RX) Децентрализованный выход (RY)	<ul style="list-style-type: none"> <li>Старший байт Адрес децентрализованного входа (RX) интеллектуальной станции</li> </ul>	от 0 до 124	пользователь (Адреса указывает пользователь, однако, операнды устанавливаются, сбрасываются и изменяются системой.)
			<ul style="list-style-type: none"> <li>Младший байт Адрес децентрализованного выхода (RY) интеллектуальной станции</li> </ul>	от 0 до 125	
(d2) + 1	Децентрализованный регистр (RW <sub>r</sub> )	Адрес децентрализованного регистра (RW <sub>r</sub> ) интеллектуальной станции	от 0 до 15 или FF (Указание FF означает, что не выбран никакой регистр.)		

## Переменные

Операнд	Значение			Диапазон	Устанавливает	Тип данных
d3	Битовый операнд, устанавливаемый на один цикл после выполнения команды RIRCV. С помощью (d3) + 1 выводится ошибка при выполнении.					
	Операнд	Значение	Описание	Диапазон	Устанавливает	бит
	(d3) + 0	Команда выполнена	Показывает завершение команды. Вкл.: команда выполнена Выкл.: команда не выполнена	—	система	
(d3) + 1	команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды. Вкл.: команда выполнена с ошибкой Выкл.: команда выполнена без ошибок	—			

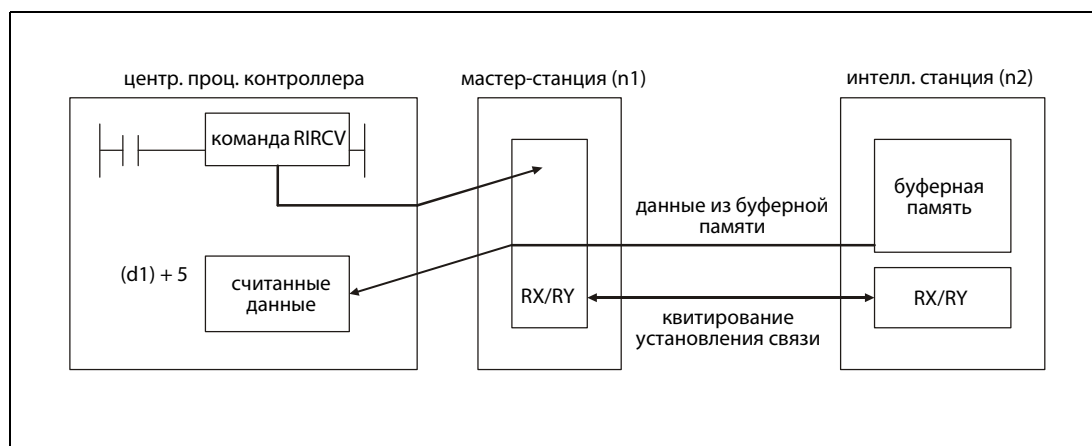
## Принцип действия

### Считывание данных из буферной памяти интеллектуальной станции с квитированием RIRCV      Считывание данных (с квитированием)

Выполнение команды RIRCV возможно только в центральном процессоре контроллера мастер-станции. Эта команда служит для считывания данных из буферной памяти интеллектуальной станции по сети CC-Link. Обмен данными происходит с квитированием (handshake).

В операнде (d1) + 1 указывается, сколько данных требуется передать. В операнде (d1) + 3 указывается первый адрес в буферной памяти, которую требуется считать. Операнд n2 содержит номер другой станции. Эта станция подключена к мастер-станции, указанной в n1. Считанные данные сохраняются в центральном процессоре, обрабатывающем команду RIRCV, начиная с (d1) + 5.

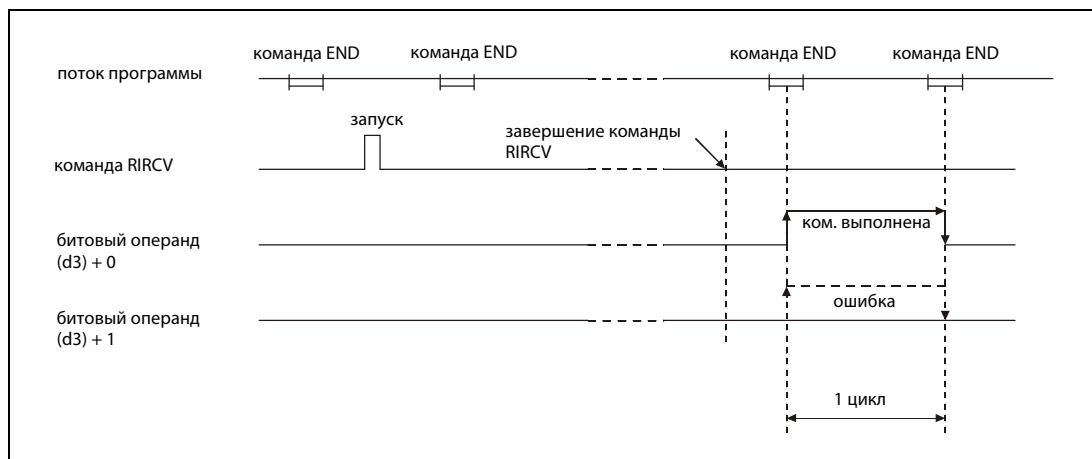
Функциональная схема команды RIRCV:



Операнды, указанные в (d3) + 0 и (d3) + 1, показывают, выполнена ли команда и возникла ли при этом ошибка.

- Битовый операнд (d3) + 0 устанавливается при выполнении команды END того цикла, в котором была завершена команда RIRCV. При обработке следующей команды END этот битовый операнд снова сбрасывается.
- Битовый операнд (d3) + 1 показывает ошибку при выполнении команды RIRCV. При безошибочном выполнении этот битовый операнд остается сброшенным. При наличии ошибки операнд (d3) + 1 устанавливается при выполнении команды END того цикла, в котором была завершена команда RIRCV. При обработке следующей команды END этот операнд снова сбрасывается.

На рисунке ниже показана диаграмма изменения сигналов при выполнении команды RIRCV:



Команду RIRCV можно выполнить одновременно для нескольких станций. Однако к интеллектуальной станции невозможно одновременно обращаться несколькими командами RIRCV из различных станций.

#### Условия выполнения

Если команда RIRCV запрограммирована в сочетании с командой LEDA, то команда RIRCV выполняется до тех пор, пока условие выполнения команды LEDA включено.

При применении команды LEDB процесс считывания выполняется только при положительном фронте условия выполнения.

Учитывайте, что для обработки команды RIRCV нужны несколько циклов. Поэтому запускайте очередной процесс считывания лишь после того, как операнд (d3) + 0 показал завершение обработки команды RIRCV. (Если команда RIRCV снова запущена прежде, чем была завершена обработка предыдущей команды RIRCV, новая команда не выполняется.)

#### Источники ошибок

Перед выполнением команды RIRCV параметры сети должны быть установлены с помощью команды RLPA. Если это не было сделано, то после выполнения команды RIRCV в операнд (d1) + 0 записывается код ошибки 4B00<sub>H</sub>.

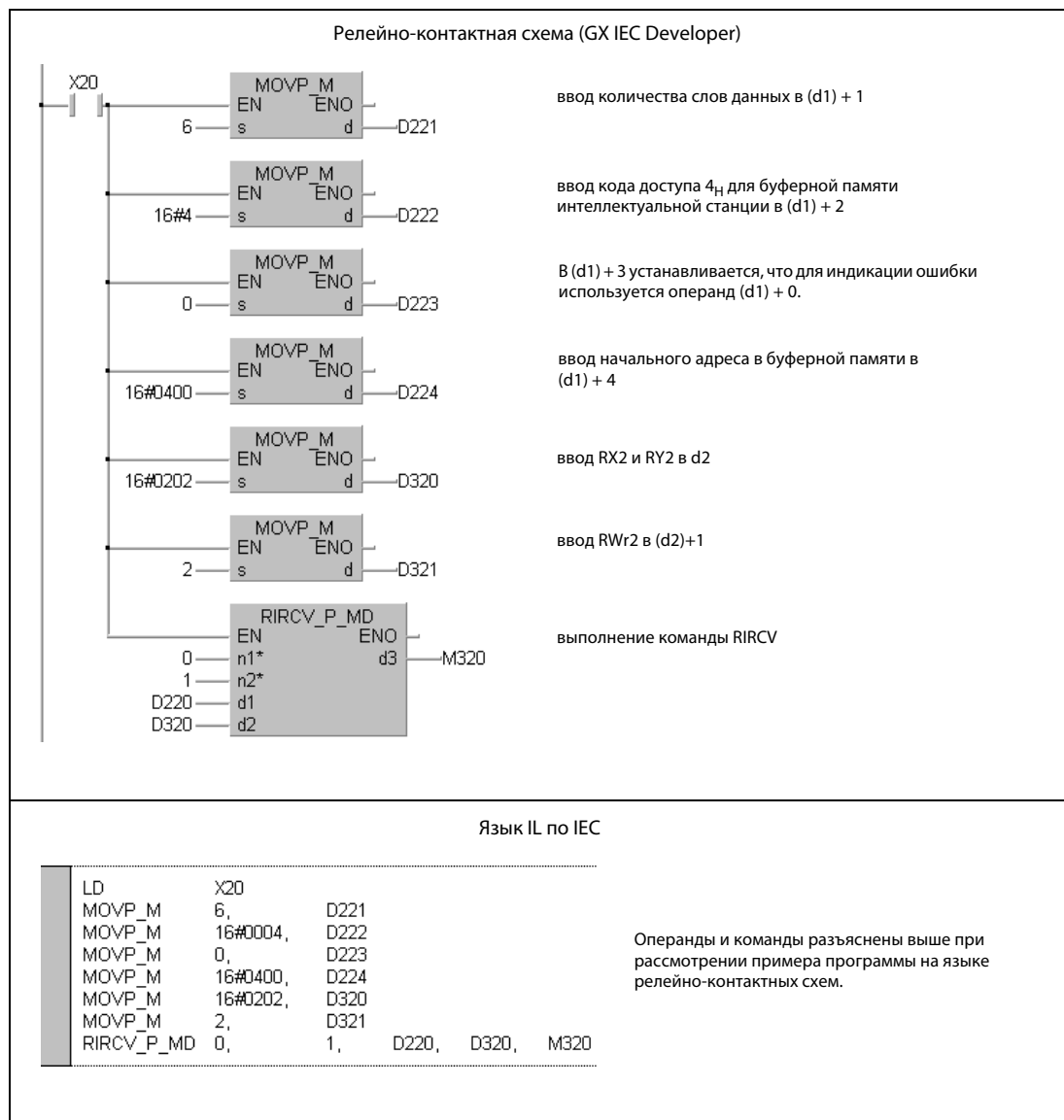
Если количество считываемых данных, указанное в (d1) + 1, равно 0 или находится вне диапазона от 1 до 480, то при завершении команды RIRCV в операнд (d1) + 0 записывается код ошибки BB42<sub>H</sub>.

**Пример**

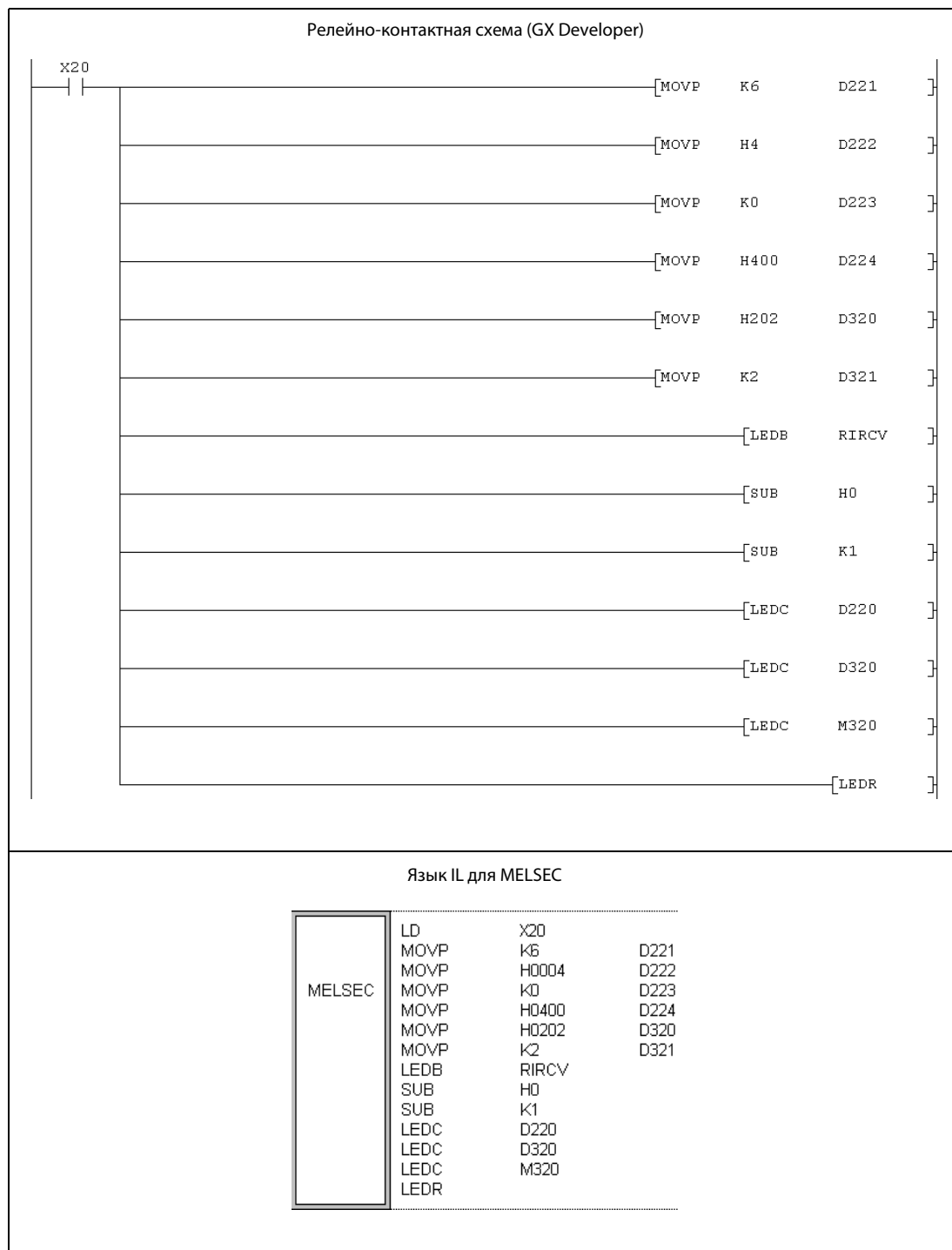
RIRCV

Следующая программа, обрабатываемая контроллером мастер-станции, считывает содержимое адресов 400<sub>H</sub>...405<sub>H</sub> буферной памяти интеллектуальной станции № 1. В качестве операндов для квитирования связи используются регистры RX2, RY2 и RWr2. Операнд, указанный в (d1) + 0, служит для индикации ошибки. Головной адрес модуля CC-Link мастер-станции равен X/Y000.

- Редакторы по стандарту IEC (на следующей странице этот пример воспроизведен на языке списка инструкций MELSEC и языке LD в среде GX Developer)



- Список инструкций MELSEC и релейно-контактная схема в среде GX Developer  
Операнды и команды разъяснены на предыдущей странице при рассмотрении примера программы на языке LD в среде GX IEC Developer.



Указания по программированию специализированных команд для контроллера серии "A" содержатся в разделе 3.3 этого руководства.

### 11.5.9 RIRCV (серии QnA и "System Q")

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

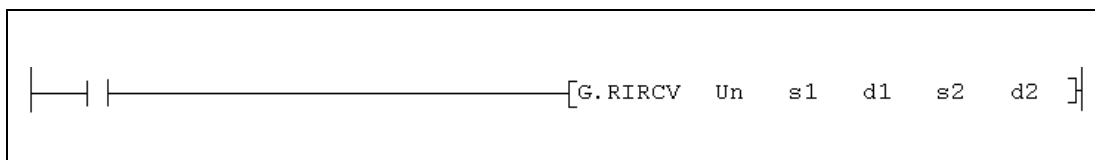
Операнды MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., польз.ов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□□□		Спец. модули U□V□□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	—	SM0	10
s1	—	●	●	—	—	—	—	—	—		
d1	—	●	●	—	—	—	—	—	—		
d2	●	●	●	—	—	—	—	—	—		

GX IEC Developer



GX Developer



## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных	
Un	Головной адрес мастер-модуля на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде 10 <sub>H</sub> )	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит	
s1	Первый операнд области, содержащей информацию для выполнения команды.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(s1) + 0	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды: 0000 <sub>H</sub> : безошибочная обработка Любое иное значение кроме 0000 <sub>H</sub> : при обработке возникла ошибка (см. руководство по модулю CC-Link).	—	система
	(s1) + 1	Номер станции	Номер интеллектуальной станции, из которой требуется считать данные.	от 0 до 64	пользователь
	(s1) + 2	Код доступа	Укажите значение 0004 <sub>H</sub> (считывание из буферной памяти интеллектуальной станции).	0004 <sub>H</sub>	
	(s1) + 3	Начальный адрес	Начальный адрес в буферной памяти	В завис. от станции, к которой осуществл. доступ	
(s1) + 4	Длина данных	Укажите здесь, сколько данных (слов) требуется считать. Указанное значение должно находиться в пределах длины буферной памяти интеллектуальной станции и размера принимающего буфера мастер-станции.	от 1 до 480		
s2	Указание операндов связи, используемых для квитирования связи.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(s2) + 0	Децентрализованный выход (RY) для запроса данных	● Старший байт Устанавливается на "0".	0	пользователь
			● Младший байт Адрес децентрализованного выхода (RY) интеллектуальной станции	от 0 до 127	
	(s2) + 1	Децентрализованный регистр (RW <sub>r</sub> ) для кода ошибки Децентрализованный вход (RX), используемый для индикации завершения передачи данных.	● Старший байт Адрес децентрализованного регистра (RW <sub>r</sub> ) интеллектуальной станции, в который записывается тот же код ошибки, что и в операнд, указанный в (s1) + 0	от 0 до 15 или FF (Указание FF означает, что не выбран никакой регистр.)	пользователь
● Младший байт Адрес децентрализованного входа (RX) интеллектуальной станции			от 0 до 127		
(s2) + 2	Индикация завершения передачи данных	Выберите, каким образом отображается завершение передачи данных: 0: с помощью одного операнда (RX <sub>n</sub> ) 1: с помощью 2 операндов (RX <sub>n</sub> , RX <sub>n</sub> + 1) (RX <sub>n</sub> + 1 устанавливается при наличии ошибки.)	0 или 1	BIN, 16 бит	



## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных	
d1	Первый адрес области операндов, в которой сохраняются считанные данные.		пользователь	BIN, 16 бит	
d2	Битовый операнд, устанавливаемый на один цикл после выполнения команды RIRCV. По адресу (d2) + 1 выводится ошибка при выполнении.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(d2) + 0	Команда выполнена	Показывает завершение команды RIRCV. ВКЛ.: команда выполнена ВЫКЛ.: команда не выполнена	—	система
(d2) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды RIRCV. ВКЛ.: команда выполнена с ошибкой ВЫКЛ.: команда выполнена без ошибок	—		

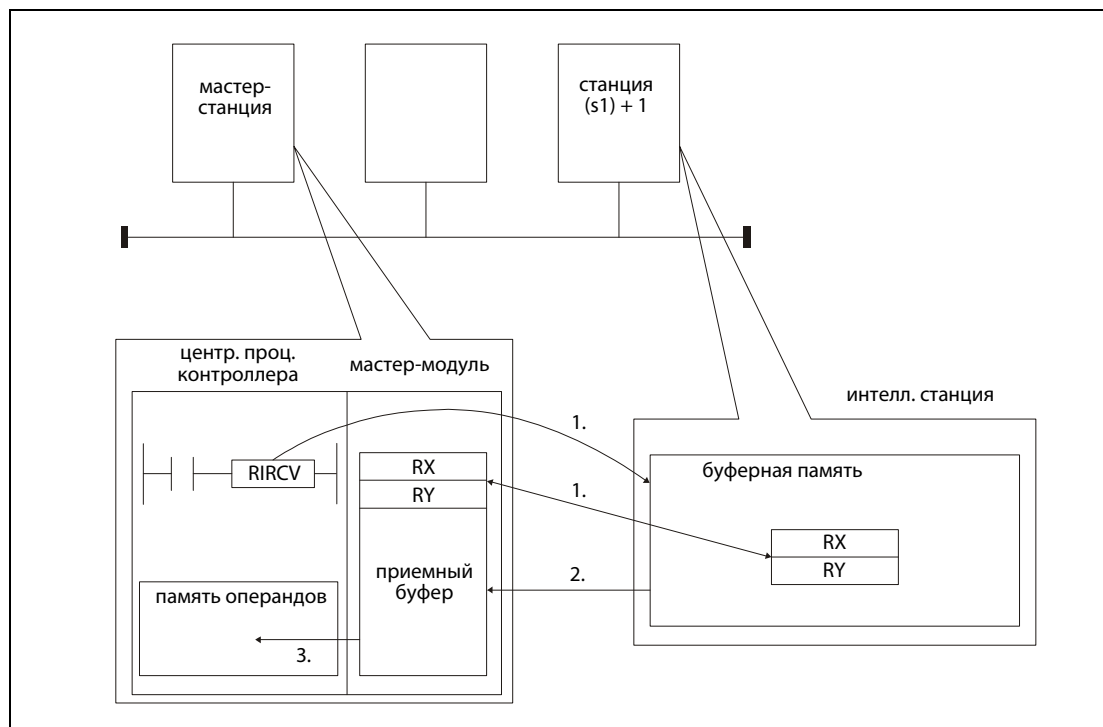
## Принцип действия

## Считывание данных из буферной памяти интел. станции с квитированием (handshake)

## RIRCV Считывание данных (с квитированием)

Выполнение команды RIRCV возможно только в центральном процессоре контроллера мастер-станции. Эта команда служит для считывания данных из буферной памяти интеллектуальной станции по сети CC-Link. Обмен данными происходит с квитированием (handshake).

Функциональная схема команды RIRCV:

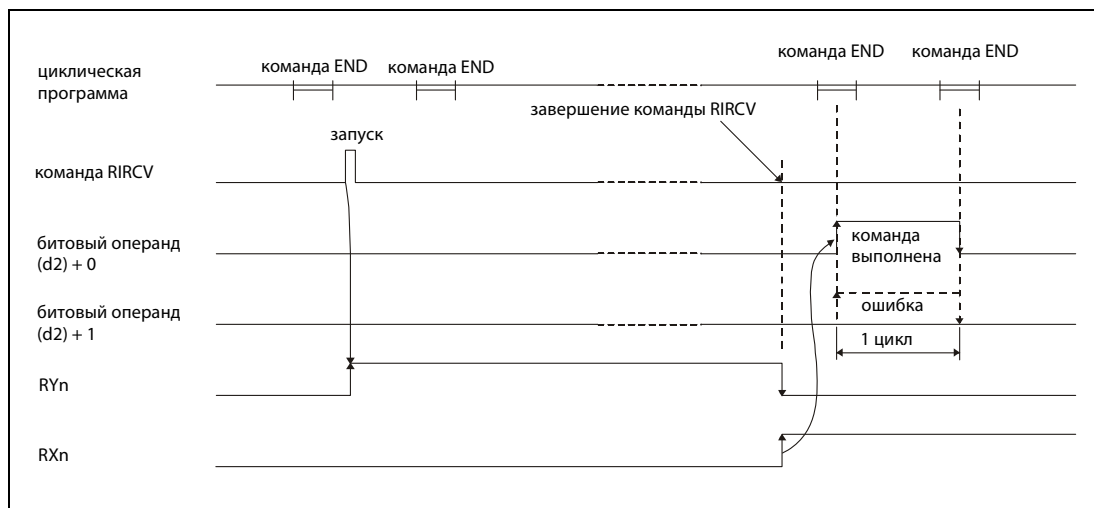


1. Доступ осуществляется к буферной памяти станции, указанной в (s1) + 1, начиная с адреса, указанного в (s1) + 3. Операнд, указанный в s2, используется для квитирования связи.
2. Содержимое адресов буферной памяти записывается в приемный буфер мастер-модуля.
3. В центральном процессоре контроллера мастер-станции данные сохраняются, начиная с операнда, указанного в d1. Затем устанавливается битовый операнд, указанный в (d2) + 0.

Операнды в (d2) + 0 и (d2) + 1 показывают, выполнена ли команда и возникла ли при ее выполнении ошибка.

- Битовый операнд (d2) + 0 устанавливается при выполнении команды END того цикла, в котором команда RIRCV была завершена. При обработке следующей команды END этот битовый операнд снова сбрасывается.
- Битовый операнд (d2) + 1 показывает ошибку при выполнении команды RIRCV. При безошибочном выполнении этот битовый операнд остается сброшенным. Если имеется ошибка, то операнд (d2) + 1 устанавливается при выполнении команды END того цикла, в котором команда RIRCV была завершена. При обработке следующей команды END этот операнд снова сбрасывается.

На рисунке ниже показана диаграмма изменения сигналов при выполнении команды RIRCV:



Команду RIRCV можно выполнить одновременно для нескольких станций. Однако к интеллектуальной станции невозможно одновременно обращаться несколькими командами RIRCV из различных станций.

#### Источники ошибок

Если при выполнении команды RIRCV возникла ошибка, то устанавливается флаг ошибки SMO и в специальный регистр SD0 записывается код ошибки.

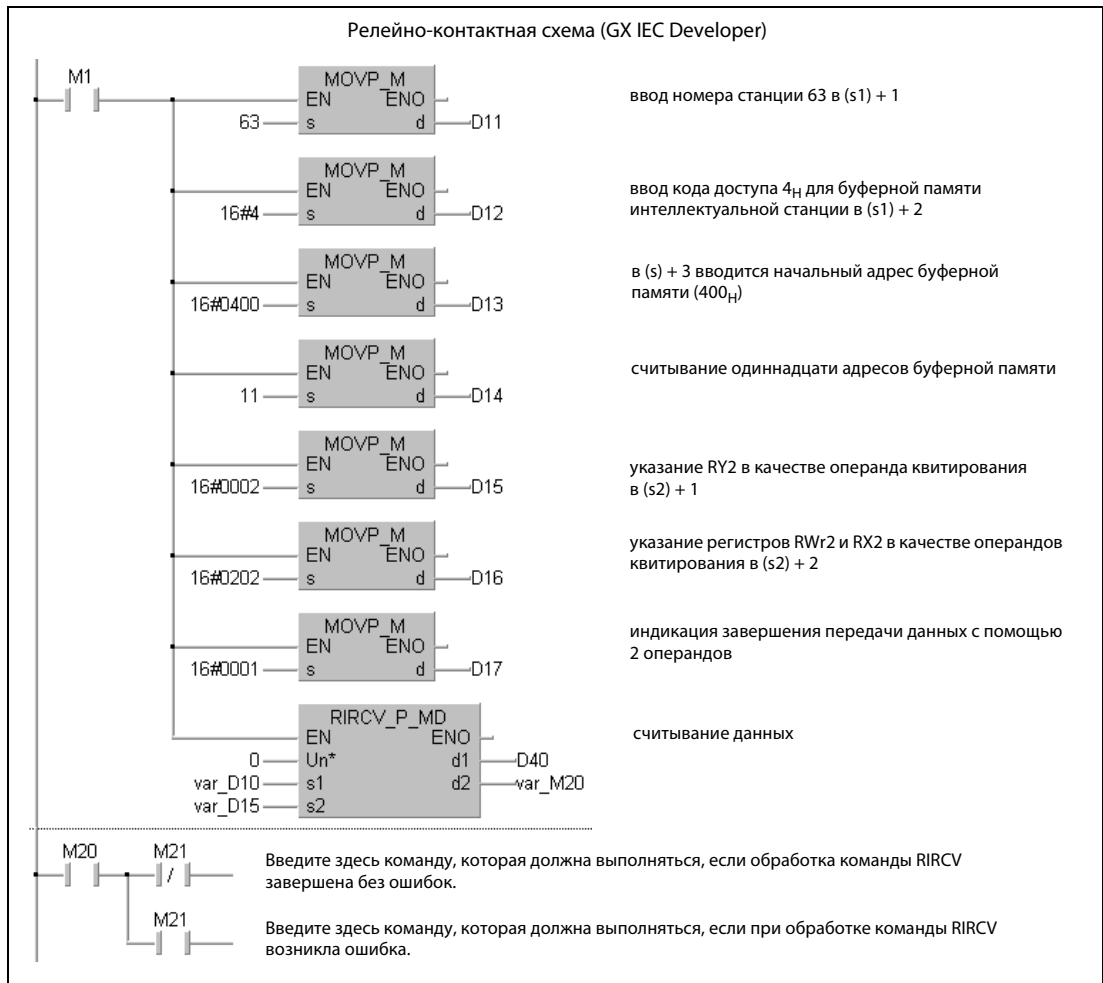
- Указанный в Un модуль не является специальным модулем (код ошибки: 2112).
- Выполняемая команда не поддерживается (код ошибки: 4002).
- В команде указано недействительное количество операндов (код ошибки: 4003).
- В команде указан недопустимый операнд (код ошибки: 4004).
- Область операндов, определенная с помощью s, содержит недопустимые данные (код ошибки: 4100).
- Количество данных превышает допустимый диапазон (код ошибки: 4101).
- Сохраненные данные или константы, переданные с помощью этой команды, превышают допустимый диапазон (код ошибки: 4101).

**Пример**

RIRCV

Следующая программа обрабатывается в контроллере мастер-станции. Если маркер M1 установлен, то из интеллектуальной станции № 63, начиная с адреса буферной памяти 400<sub>H</sub>, считывается содержимое одиннадцати адресов буферной памяти. Эти данные сохраняются в центральном процессоре, начиная с регистра D40. Модуль CC-Link мастер-станции имеет головной адрес X/Y00. В качестве операндов для квитирования связи используются регистры RX2, RY2 и RWr2. Завершение передачи данных обозначается двумя операндами. ((s2) + 2 = 1).

- Редакторы по стандарту IEC (на следующей странице этот пример воспроизведен на языке списка инструкций MELSEC и языке LD в среде GX Developer)



Язык IL по IEC

```

LD      M1
MOV_P_M 63, D11
MOV_P_M 16#0004, D12
MOV_P_M 16#0400, D13
MOV_P_M 11, D14
MOV_P_M 16#0002, D15
MOV_P_M 16#0202, D16
MOV_P_M 16#0001, D17
RIRCV_P_MD 0, var_D10, var_d15, D40, var_M20
    
```

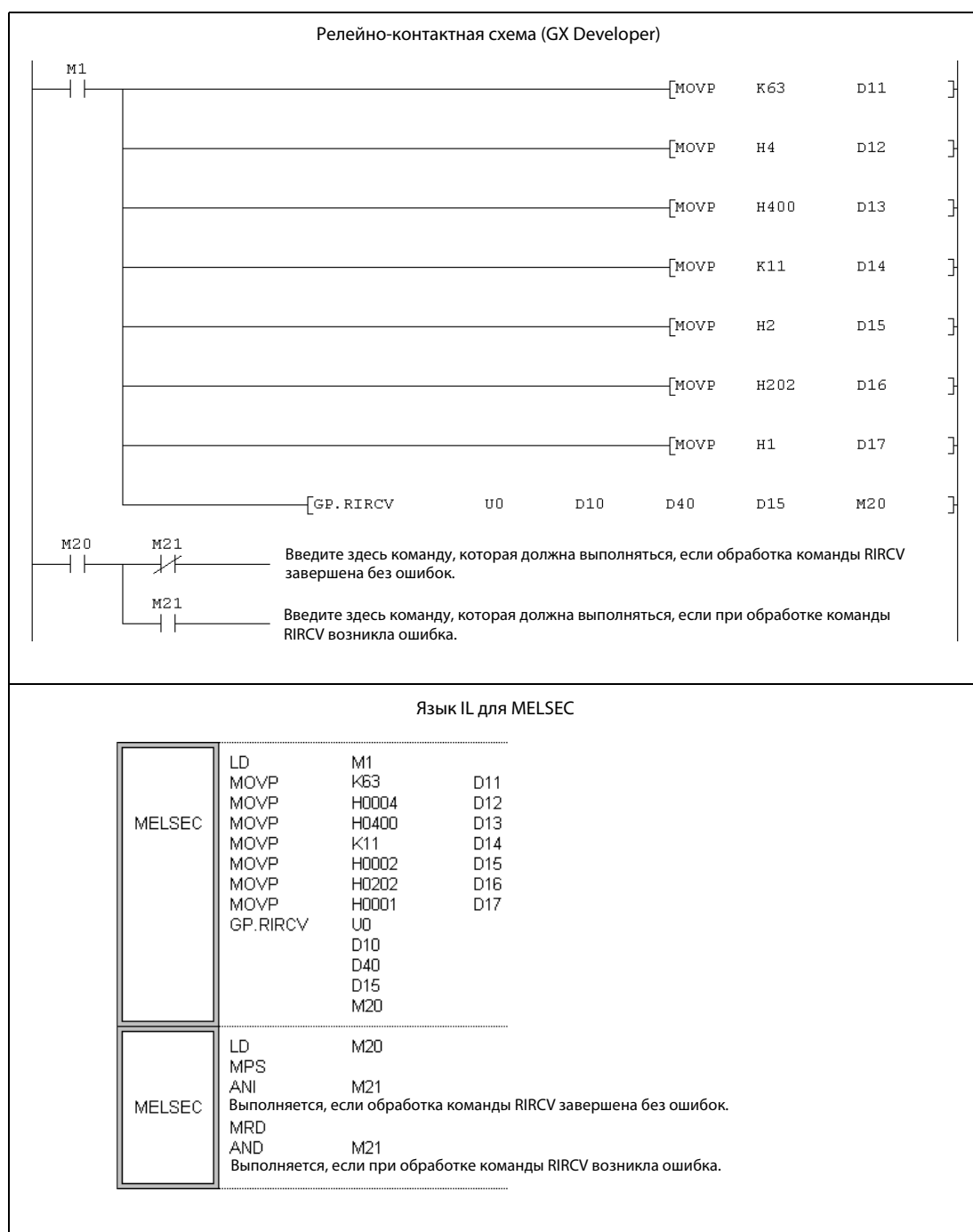
Операнды и команды разъяснены выше при рассмотрении примера программы на языке релейно-контактных схем.

```

LD      M20
ANDN   M21
      Выполняется, если обработка команды RIRCV завершена без ошибок.
LD      M20
AND     M21
      Выполняется, если при обработке команды RIRCV возникла ошибка.
    
```

**ПРИМЕЧАНИЕ** В редакторах по стандарту IEC среды GX IEC Developer переменные должны быть объявлены в заголовке программного компонента (POU). Если переменные не были объявлены, то при проверке или компилировании программы возникает сообщение об ошибке.  
Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

- Список инструкций MELSEC и релейно-контактная схема в среде GX Developer  
Операнды и команды разъяснены на предыдущей странице при рассмотрении примера программы на языке LD в среде GX IEC Developer.



### 11.5.10 RISEND (серия "A")

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

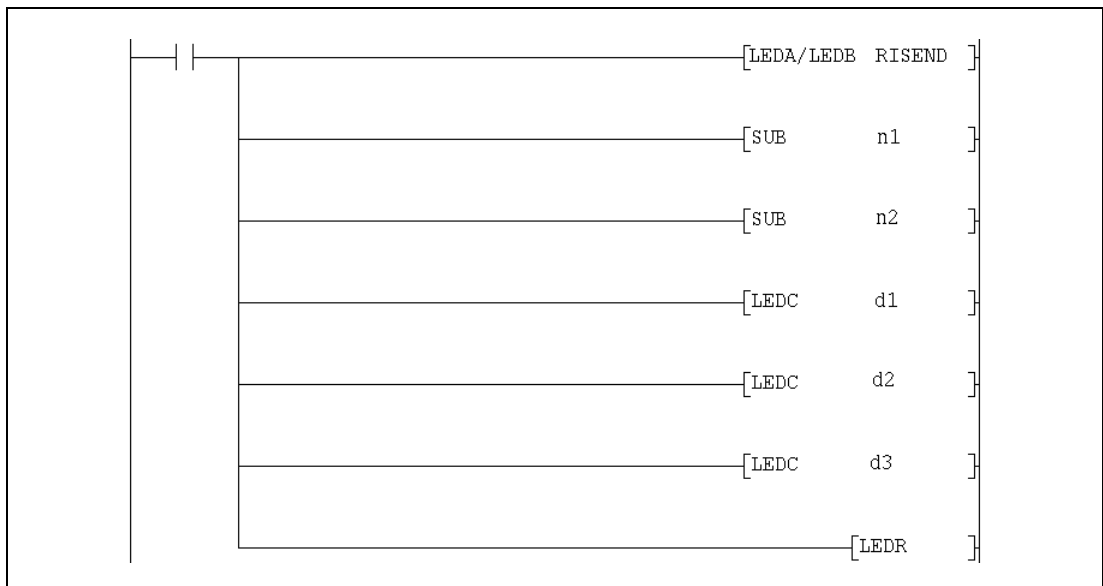
Операнды MELSEC A

	Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9011	
	битовые						словные (16 бит)						Константы		Указатели							Уровень
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						
n1																	●	●				
n2																	●	●				
d1							●	●	●	●	●											
d2							●	●	●	●	●											
d3	●	●	●	●	●																	

GX IEC Developer



GX Developer



## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных	
n1	Головной адрес мастер-модуля CC-Link на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде 10 <sub>H</sub> )	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит	
n2	Номер интеллектуальной станции, в которую требуется передать данные.	от 1 до 64	пользователь	BIN, 16 бит	
d1	Первый операнд области, содержащей информацию для выполнения команды.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(d1) + 0	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды: 0000 <sub>H</sub> : безошибочная обработка Любое иное значение кроме 0000 <sub>H</sub> : при обработке возникла ошибка. Более подробные указания, относящиеся к кодам ошибок, имеются в руководстве по эксплуатации модуля CC-Link	—	система
	(d1) + 1	Длина данных	Укажите здесь, сколько данных (слов) требуется записать.	от 1 до 480	пользователь
	(d1) + 2	Код доступа	Укажите значение 0004 <sub>H</sub> . (= запись в буферную память интеллектуальной станции)	0004 <sub>H</sub>	
	(d1) + 3	Выявление ошибок	Укажите операнд, используемый для индикации ошибок при выполнении команды RISEND: 0: операнд d1 1: операнд RX + 1	0 или 1	
	(d1) + 4	Начальный адрес	Начальный адрес в буферной памяти интеллектуальной станции	В завис. от станции, к которой осуществл. доступ.	
от (d1) + 5 до (d1) + n	Область памяти для передаваемых данных	Размер этой области определяется длиной данных, указанной в (d1) + 1.	—		
d2	Указание операндов связи, используемых для квитирования связи.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(d2) + 0	Децентрализованный вход (RX) Децентрализованный выход (RY)	● Старший байт Адрес децентрализованного входа (RX) интеллектуальной станции	от 0 до 127	пользователь (Адреса указывает пользователь, однако, операнды устанавливает, сбрасывает и изменяет система.)
● Младший байт Адрес децентрализованного выхода (RY) интеллектуальной станции			от 0 до 127		
(d2) + 1	Децентрализованный регистр (RW <sub>r</sub> )	Адрес децентрализованного регистра (RW <sub>r</sub> ) интеллектуальной станции	от 0 до 15 или FF (Указание FF означает, что не выбран никакой регистр.)		

## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных	
d3	Битовый операнд, устанавливаемый на один цикл после выполнения команды RISEND. С помощью (d3) + 1 выводится ошибка при выполнении.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(d3) + 0	Команда выполнена	Показывает завершение команды. Вкл.: команда выполнена Выкл.: команда не выполнена	—	система
(d3) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды. Вкл.: команда выполнена с ошибкой Выкл.: команда выполнена без ошибок	—		
				бит	

## Принцип действия

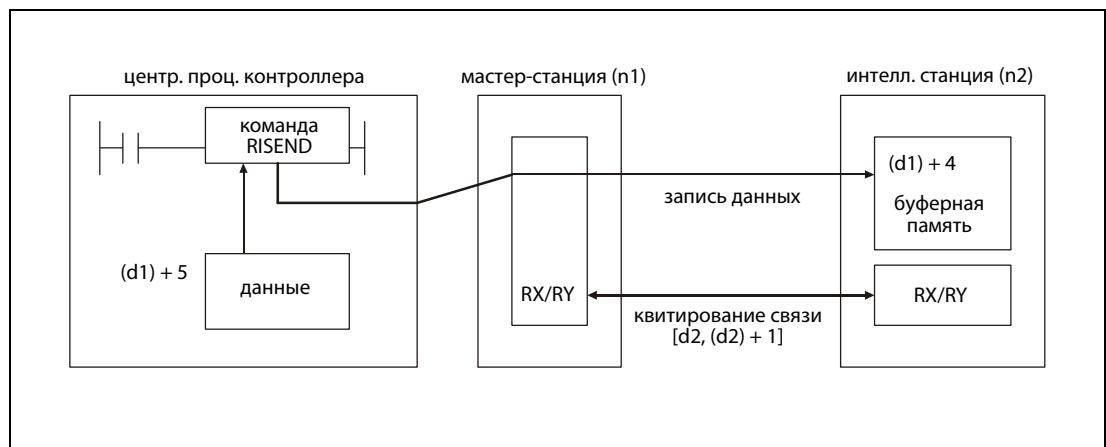
## Передача данных в буферную память интеллектуальной станции с квитированием (handshake)

## RISEND Запись данных (с квитированием)

Команда RISEND может быть выполнена только в центральном процессоре контроллера мастер-станции. С ее помощью данные передаются в буферную память интеллектуальной станции CC-Link. Обмен данными происходит с квитированием (handshake).

В операнде (d1) + 1 указывается, сколько данных требуется передать. В операнде (d1) + 3 указывается первый адрес в буферной памяти, в который требуется передать данные. Операнд n2 содержит номер другой станции. Эта станция подключена к мастер-станции, указанной в n1. Данные, передаваемые в интеллектуальную станцию, хранятся, начиная с операнда (d1) + 5.

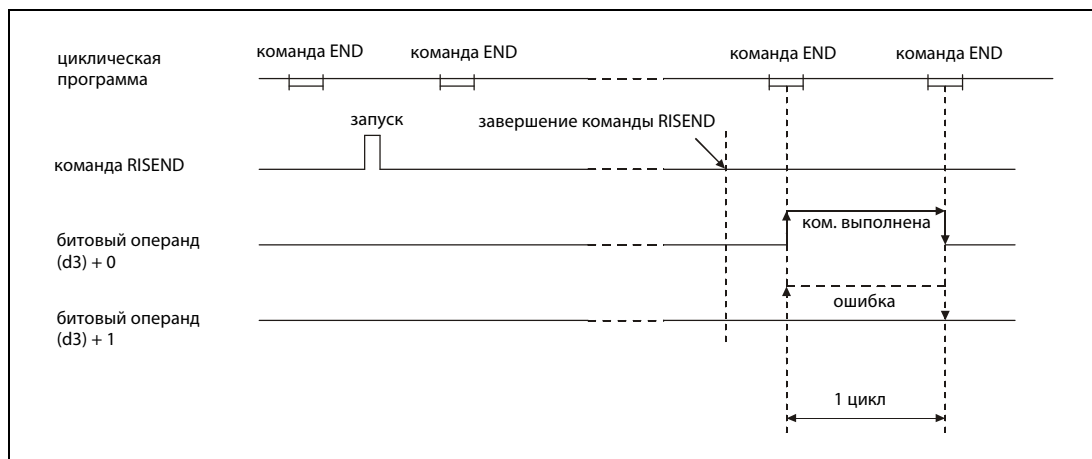
Функциональная схема команды RISEND:



Операнды, указанные в (d3) + 0 и (d3) + 1, показывают, выполнена ли команда и возникла ли при этом ошибка.

- Битовый операнд (d3) + 0 устанавливается при выполнении команды END того цикла, в котором была завершена команда RISEND. При обработке следующей команды END этот битовый операнд снова сбрасывается.
- Битовый операнд (d3) + 1 показывает ошибку при выполнении команды RISEND. При безошибочном выполнении этот битовый операнд остается сброшенным. При наличии ошибки операнд (d3) + 1 устанавливается при выполнении команды END того цикла, в котором была завершена команда RISEND. При обработке следующей команды END этот операнд снова сбрасывается.

На рисунке ниже показана диаграмма изменения сигналов при выполнении команды RISEND:



Команду RISEND можно выполнить одновременно для нескольких станций. Однако к одной интеллектуальной станции невозможно одновременно обращаться с помощью нескольких команд RISEND из различных станций.

#### Условия выполнения

Если команда RISEND запрограммирована в сочетании с командой LEDA, то команда RISEND выполняется до тех пор, пока условие выполнения команды LEDA включено.

При применении команды LEDB процесс считывания выполняется только при положительном фронте условия выполнения.

Учитывайте, что для обработки команды RISEND нужны несколько циклов. Поэтому запускайте очередной процесс считывания лишь после того, как операнд (d3) + 0 показал завершение обработки команды RISEND. (Если команда RISEND снова запущена прежде, чем была завершена обработка предыдущей команды, новая команда не выполняется.)

Перед выполнением команды RISEND должны быть установлены параметры сети с помощью команды RLPA.

#### Источники ошибок

Если количество передаваемых данных, указанное в (d1) + 1, равно 0 или находится вне диапазона от 1 до 480, то при завершении команды RISEND в операнд (d1) + 0 записывается код ошибки ВВ42<sub>H</sub>.

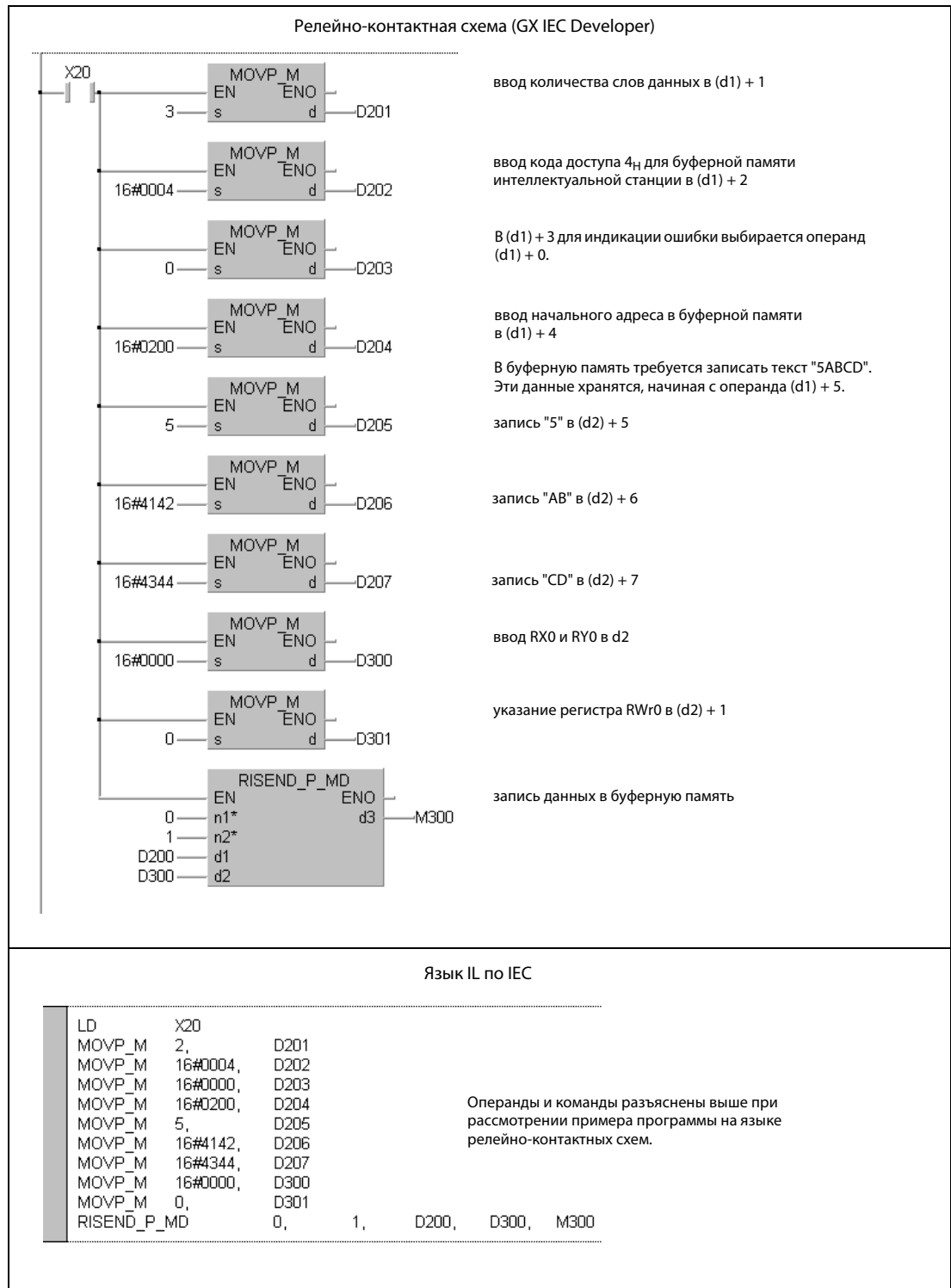


**Пример**

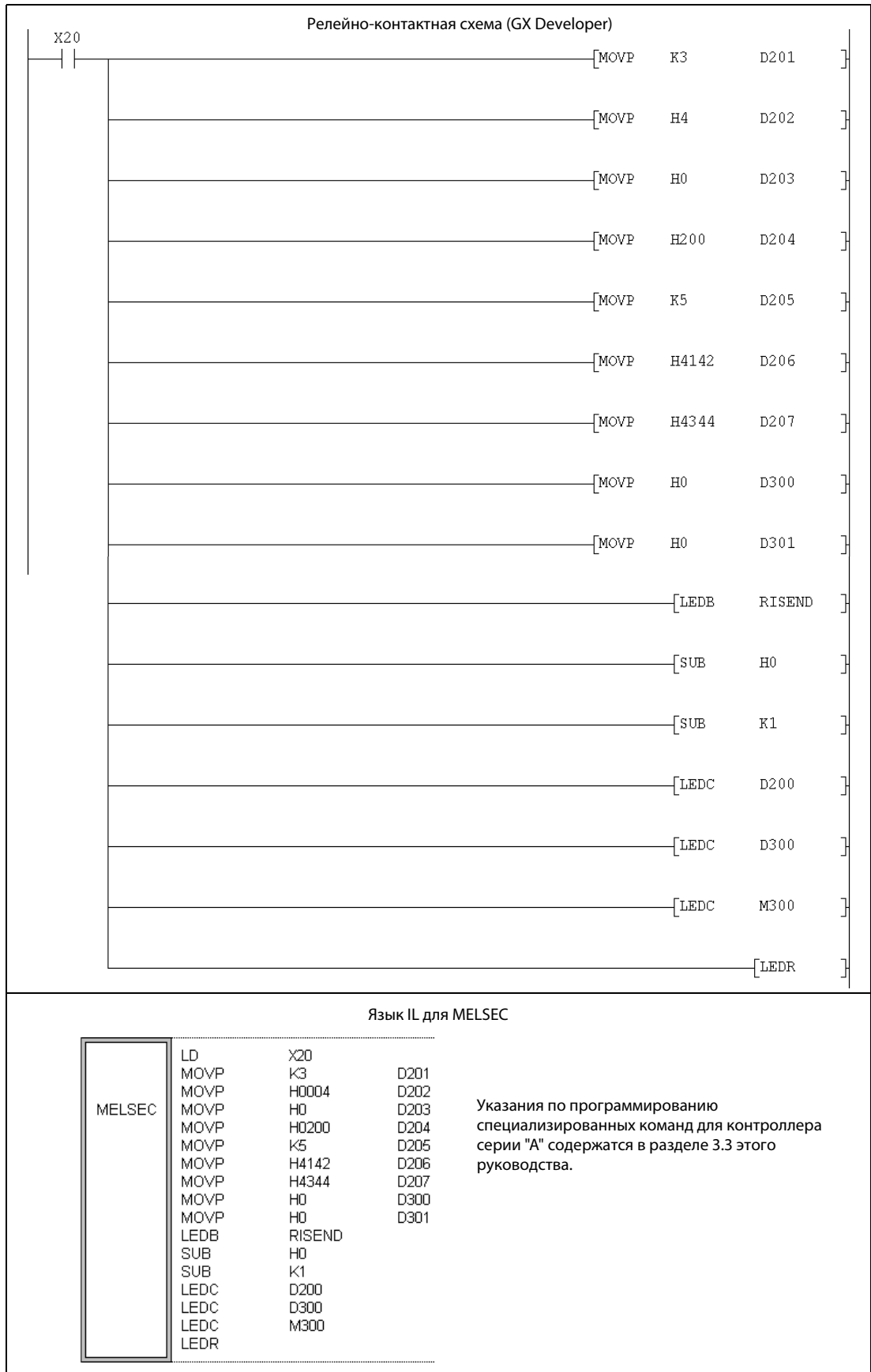
**RISEND**

Следующая программа обрабатывается в контроллере мастер-станции и записывает данные в адреса 200<sub>H</sub>...202<sub>H</sub> буферной памяти интеллектуальной станции № 1. В качестве операндов для квитирования связи используются регистры RX0, RY0 и RWr0. Операнд, указанный в (d1) + 0, служит для индикации ошибки. Головной адрес модуля CC-Link мастер-станции равен X/Y000.

- Редакторы по стандарту IEC (на следующей странице этот пример воспроизведен на языке списка инструкций MELSEC и языке LD в среде GX Developer)



- Список инструкций MELSEC и релейно-контактная схема в среде GX Developer  
 Операнды и команды разъяснены на предыдущей странице при рассмотрении примера программы на языке LD в среде GX IEC Developer.



### 11.5.11 RISEND (серии QnA и "System Q")

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

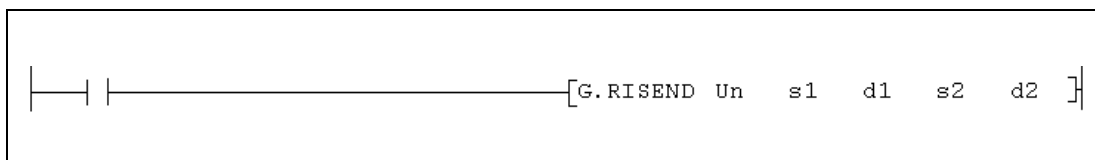
**Операнды MELSEC Q**

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
s1	—	●	●	—	—	—	—	—	SM0	10	
s2	—	●	●	—	—	—	—	—			
d1	—	●	●	—	—	—	—	—			
d2	●	●	●	—	—	—	—	—			

**GX IEC Developer**



**GX Developer**



## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных	
Un	Головной адрес мастер-модуля на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде 10 <sub>H</sub> )	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит	
s1	Первый операнд области, содержащей информацию для выполнения команды.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(s1) + 0	Состояние выполнения команды	Показывает, возникла ли ошибка при обработке команды: 0000 <sub>H</sub> : безошибочная обработка Любое иное значение кроме 0000 <sub>H</sub> : при обработке возникла ошибка (см. руководство по модулю CC-Link)	—	система
	(s1) + 1	Номер станции	Номер интеллектуальной станции, из которой требуется считать данные.	от 0 до 64	пользователь
	(s1) + 2	Код доступа	Дайте значение 0004 <sub>H</sub> . (= запись в буферную память интеллектуальной станции)	0004 <sub>H</sub>	
	(s1) + 3	Начальный адрес	Начальный адрес в буферной памяти	В завис. от станции, к которой осуществл. доступ.	
(s1) + 4	Длина данных	Укажите здесь, сколько данных (слов) требуется передать.	от 1 до 480		
s2	Указание операндов связи, используемых для квитирования связи.				
	Операнд	Значение	Описание	Диапазон	Устанавливает
	(s2) + 0	Децентрализованный выход (RY) для запроса данных	● Старший байт Устанавливается на "0".	0	пользователь
			● Младший байт Адрес децентрализованного выхода (RY) интеллектуальной станции	от 0 до 127	
	(s2) + 1	Децентрализованный регистр (RWr) для кода ошибки Децентрализованный вход (RX), используемый для индикации завершения передачи данных.	● Старший байт Адрес децентрализованного регистра (RWr) интеллектуальной станции, в который записывается тот же код ошибки, что и в операнд, указанный в (s1) + 0	от 0 до 15 или FF (Указание FF означает, что не выбран никакой регистр.)	пользователь
● Младший байт Адрес децентрализованного входа (RX) интеллектуальной станции			от 0 до 127		
(s2) + 2	Индикация завершения передачи данных	Выберите, каким образом отображается завершение передачи данных: 0: с помощью одного операнда (RXn) 1: с помощью 2 операндов (RXn, RXn + 1) (RXn + 1 устанавливается при наличии ошибки.)	0 или 1		
d1	Первый адрес области операндов, в котором сохранены данные, передаваемые в интеллектуальную станцию.		пользователь	BIN, 16 бит	

## Переменные

Операнд	Значение			Диапазон	Устанавливает	Тип данных
d2	Битовый операнд, устанавливаемый на один цикл после выполнения команды RISEND. По адресу (d2) + 1 выводится ошибка при выполнении.					бит
	Операнд	Значение	Описание	Диапазон	Устанавливает	
	(d2) + 0	Команда выполнена	Показывает завершение команды RISEND. ВКЛ.: команда выполнена Выкл.: команда не выполнена	—	система	
(d2) + 1	Команда выполнена с ошибкой	Показывает, возникла ли ошибка при выполнении команды RISEND. ВКЛ.: команда выполнена с ошибкой Выкл.: команда выполнена без ошибок	—			

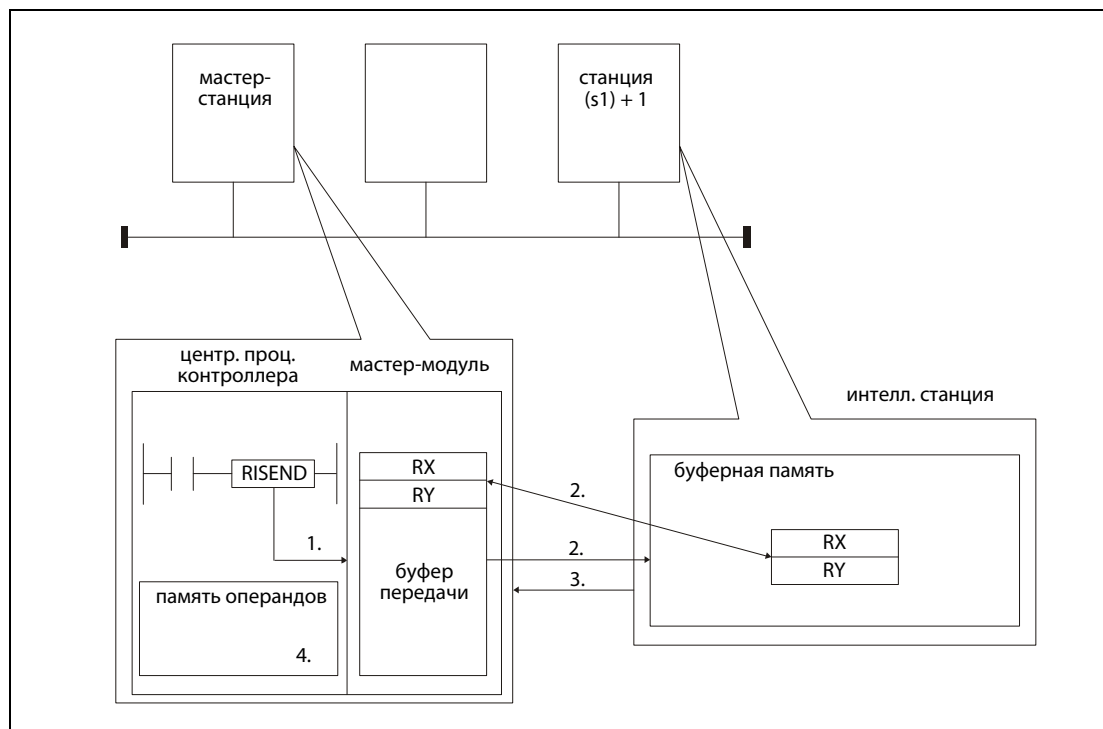
## Принцип действия

## Запись данных в буферную память интеллектуальной станции с квитированием (handshake)

## RISEND Запись данных (с квитированием)

Команда RISEND может быть выполнена только в центральном процессоре контроллера мастер-станции. С помощью этой команды данные передаются в буферную память интеллектуальной станции CC-Link. Обмен данными происходит с квитированием (handshake).

Принципиальная схема выполнения команды RISEND:

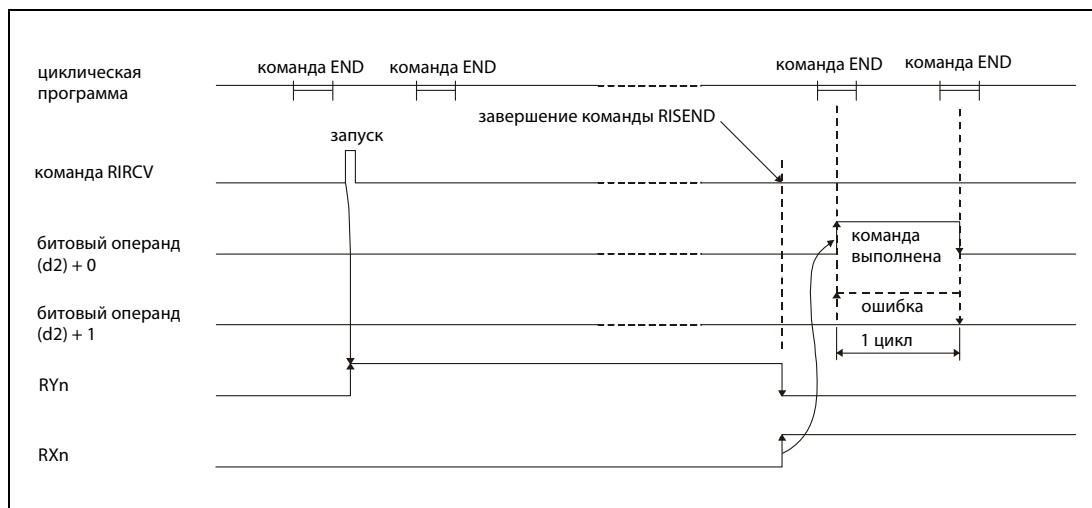


1. Данные для интеллектуальной станции вводятся в буфер передачи мастер-станции.
2. Данные записываются в буферную памяти станции, указанной в (s1) + 1, начиная с операнда, указанного в (s1) + 3. Операнды, указанные в s2, используются для квитирования связи.
3. Мастер-станция получает сигнал о том, что передача данных закончена.
4. Устанавливается битовый операнд, указанный в (d2) + 0.

Операнды в  $(d2) + 0$  и  $(d2) + 1$  показывают, выполнена ли команда и возникла ли при ее выполнении ошибка.

- Битовый операнд  $(d2) + 0$  устанавливается при выполнении команды END того цикла, в котором команда RISEND была завершена. При обработке следующей команды END этот битовый операнд снова сбрасывается.
- Битовый операнд  $(d2) + 1$  показывает ошибку при выполнении команды RISEND. При безошибочном выполнении этот битовый операнд остается сброшенным. Если имеется ошибка, то операнд  $(d2) + 1$  устанавливается при выполнении команды END того цикла, в котором команда RISEND была завершена. При обработке следующей команды END этот операнд снова сбрасывается.

На рисунке ниже показана диаграмма изменения сигналов при выполнении команды RISEND:



Команду RISEND можно выполнить одновременно для нескольких станций. Однако к одной интеллектуальной станции невозможно одновременно обращаться с помощью нескольких команд RISEND из различных станций.

### Источники ошибок

Если при выполнении команды RISEND возникла ошибка, то устанавливается флаг ошибки SMO и в специальный регистр SD0 записывается код ошибки.

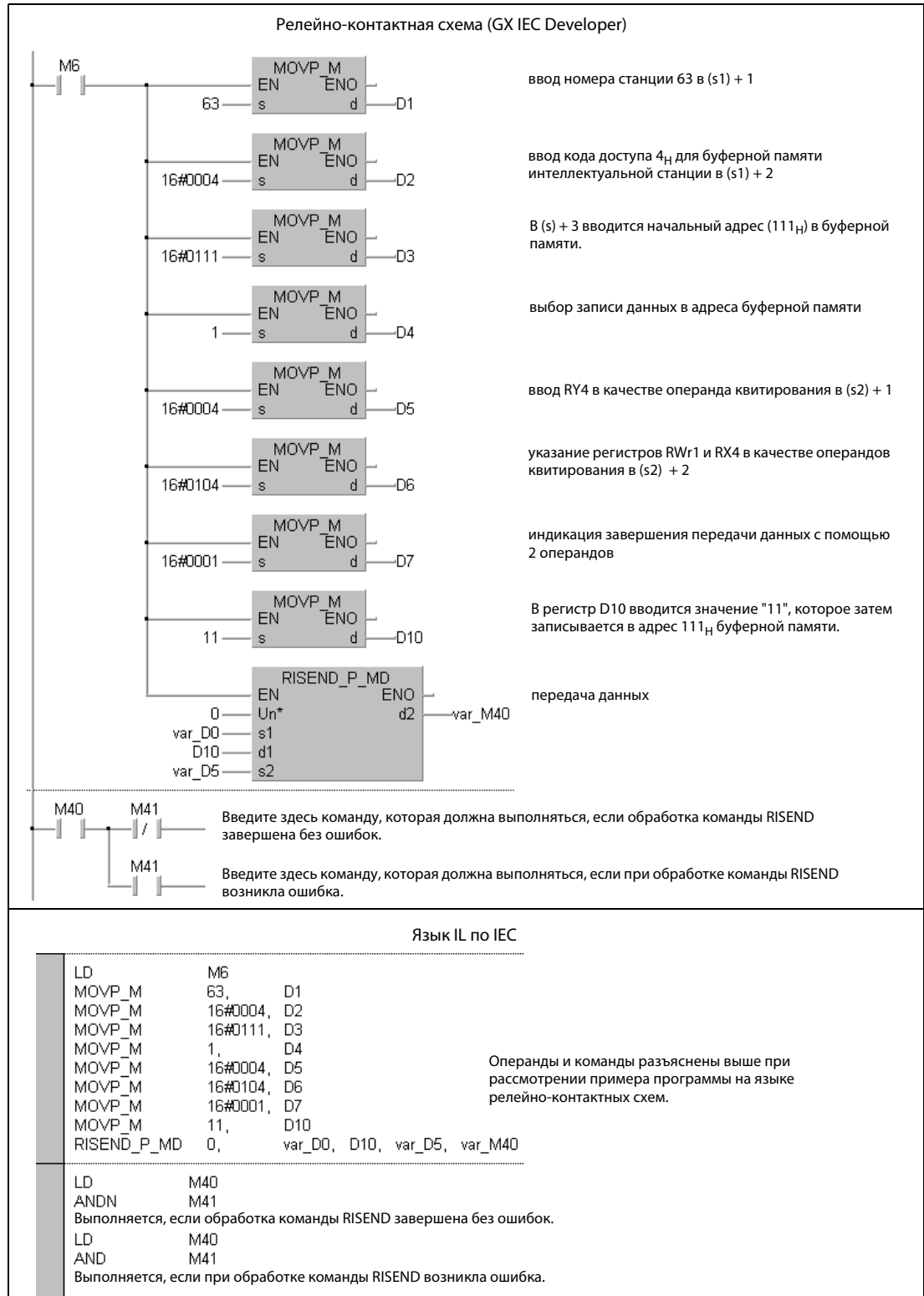
- Указанный в Un модуль не является специальным модулем (код ошибки: 2112).
- Выполняемая команда не поддерживается (код ошибки: 4002).
- В команде указано недействительное количество операндов (код ошибки: 4003).
- В команде указан недопустимый операнд (код ошибки: 4004).
- Область операндов, определенная с помощью s, содержит недопустимые данные (код ошибки: 4100).
- Количество данных превышает допустимый диапазон (код ошибки: 4101).
- Сохраненные данные или константы, переданные с помощью этой команды, превышают допустимый диапазон (код ошибки: 4101).

**Пример**

**RISEND**

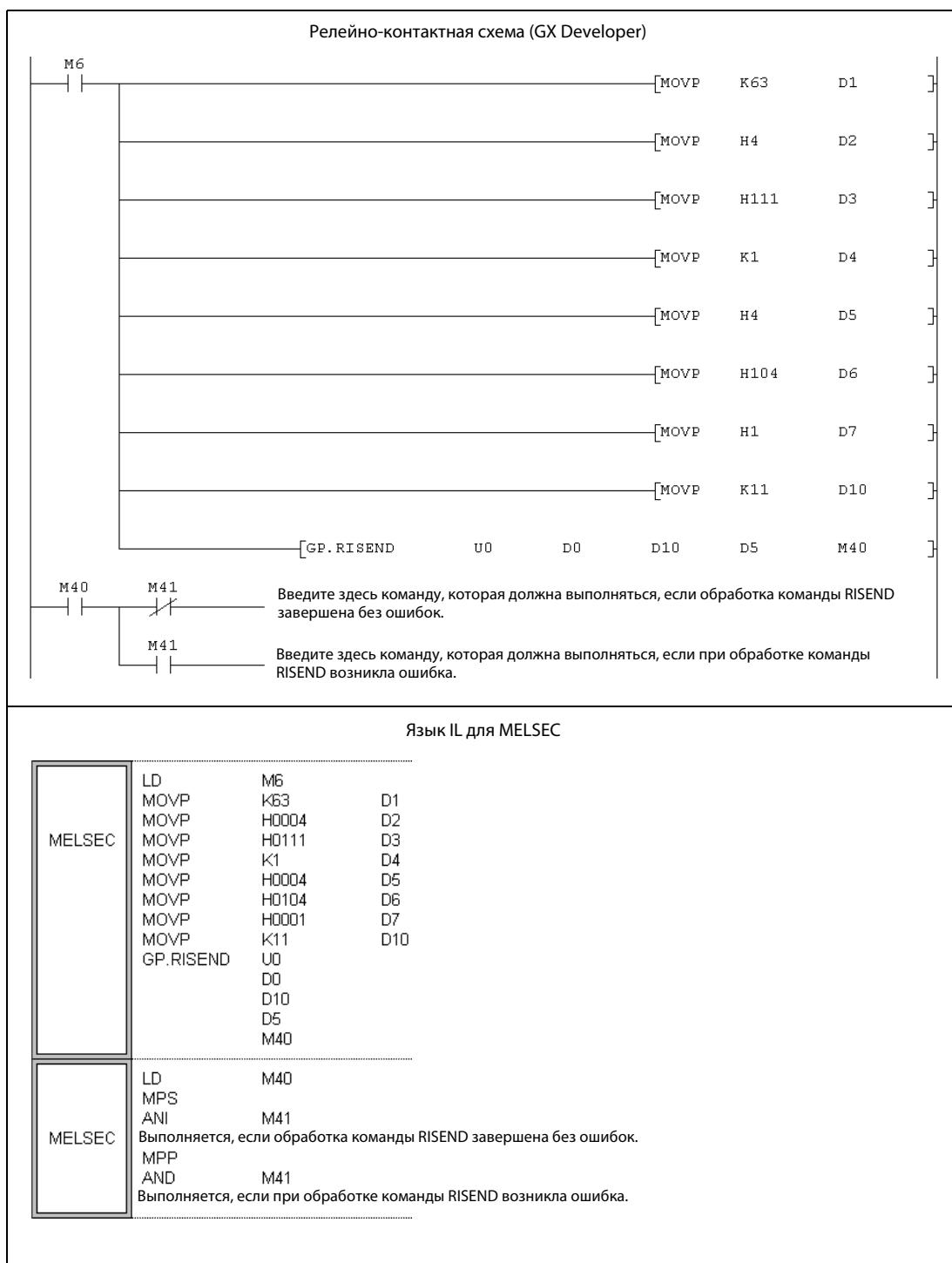
Следующая программа обрабатывается в контроллере мастер-станции и передает слово данных в адрес 111<sub>H</sub> буферной памяти интеллектуальной станции № 63. Модуль CC-Link мастер-станции имеет головной адрес X/Y00. В качестве операндов для квитирования связи используются регистры RX4, RY4 и RWr1. Завершение передачи данных обозначается двумя операндами ((s2) + 2 = 1).

- Редакторы по стандарту IEC (на следующей странице этот пример воспроизведен на языке списка инструкций MELSEC и языке LD в среде GX Developer)



**ПРИМЕЧАНИЕ** В редакторах по стандарту IEC среды GX IEC Developer переменные должны быть объявлены в заголовке программного компонента (POU). Если переменные не были объявлены, то при проверке или компилировании программы возникает сообщение об ошибке.  
Дополнительную информацию см. в разделе 3.5.2 "Адресация массивов и регистров в GX IEC Developer" этого руководства.

- Список инструкций MELSEC и релейно-контактная схема в среде GX Developer  
Операнды и команды разъяснены на предыдущей странице при рассмотрении примера программы на языке LD в среде GX IEC Developer.





### 11.5.12 RITO (серия "A")

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

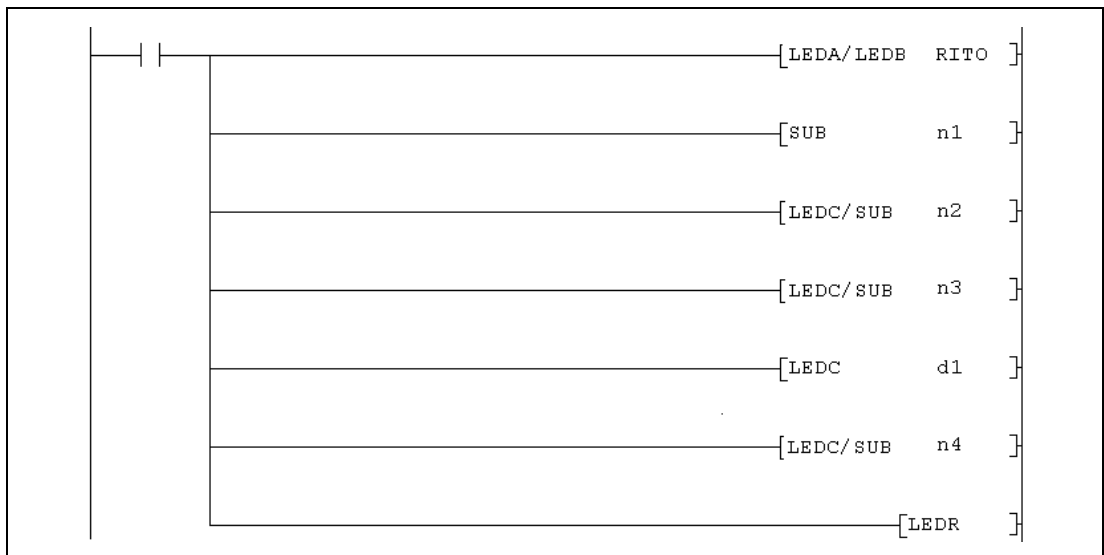
Операнды MELSEC A

	Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9011		
	битовые						словные (16 бит)						Константы		Указатели							Уровень N	
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V							K
n1																	●	●			29		
n2							●	●	●	●	●					●	●						
n3							●	●	●	●	●					●	●						
d1							●	●	●	●	●												
n4							●	●	●	●	●					●	●					●	

GX IEC Developer



GX Developer



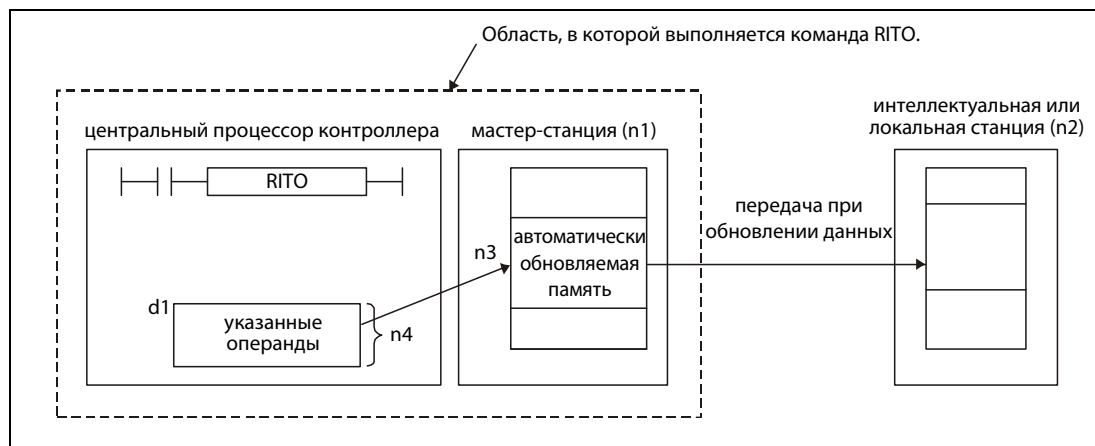
## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных
n1	Головной адрес мастер-модуля CC-Link на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде 10 <sub>H</sub> .)	от 0 до FF <sub>H</sub>	пользователь	BIN, 16 бит
n2	Цель передачи данных Укажите номер интеллектуальной станции, в которую требуется передать данные. (Только если команда RITO выполняется в мастер-станции). Если данные требуется передать в буфер со свободным доступом, укажите FF <sub>H</sub> .	от 1 до 64 или FF <sub>H</sub>		
n3	Начальный адрес автоматически обновляемой памяти для интеллектуальной станции в мастер-станции или смещение для буфера со свободным доступом	между 0 и макс. значением, установленным в параметр.		
d1	Первый операнд области, в которой сохранены передаваемые данные.	действит. адрес операнда		адрес
n4	Количество передаваемых операндов (слов)	от 1 до 4096		BIN, 16 бит

**Принцип действия****Запись данных в автоматически обновляемую память****RITO      Запись данных**

С помощью команды RITO данные из центрального процессора контроллера передаются в автоматически обновляемую буферную память мастер-станции. Данные определяются их начальным адресом (d1) и длиной (n4). Место назначения в мастер-станции указывается в n2 (номер станции, для которой предназначены данные) и n3 (начальный адрес в памяти мастер-станции). Головной адрес ввода-вывода мастер-станции указан в n1.

Функция команды RITO пояснена следующей иллюстрацией:



С помощью команды RITO можно передать максимум 4096 слов.

Для установки количества автоматически обновляемых адресов должен быть установлен размер автоматически обновляемого буфера с помощью команды RLPA.

**Условия выполнения**

Если команда RITO запрограммирована в сочетании с командой LEDA, то команда RITO выполняется до тех пор, пока условие выполнения команды LEDA включено.

При применении команды LEDB данные передаются только по при положительном фронте условия выполнения.

**Источники ошибок**

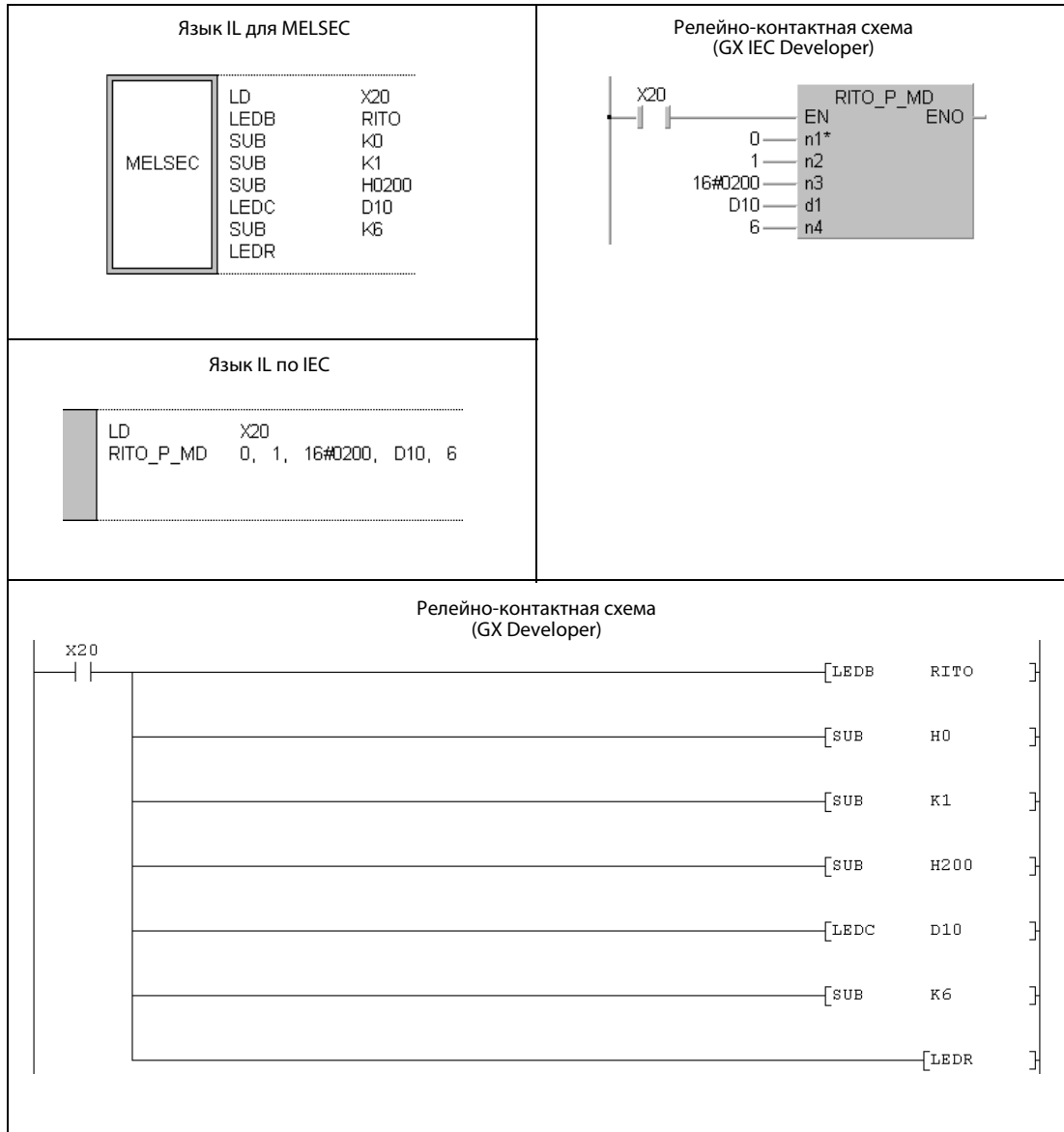
При следующих событиях распознается ошибка обработки, устанавливается флаг ошибки M9011 и выводится код ошибки.

- Настроенный адрес буферной памяти находится вне допустимого диапазона.  
(код ошибки в D9008: 50,  
код ошибки в D9091 (процессоры AnU) или D9092 (процессоры AnSH): 503)
- Количество обновляемых адресов превышает 4096.  
(код ошибки в D9008: 50,  
код ошибки в D9091 (процессоры AnU) или D9092 (процессоры AnSH): 503)

**Пример**

RITO

При установке входа X20 следующая программа передает содержимое шести регистров данных D10...D15 в автоматически обновляемую память станции № 1. В этой станции данные записываются, начиная с адреса 200H. Модуль CC-Link мастер-станции занимает диапазон адресов от X/Y000 до X/Y01F.



Указания по программированию специализированных команд в редакторах MELSEC для контроллеров серии "A" имеются в разделе 3.3 этого руководства.

### 11.5.13 RITO (серии QnA и "System Q")

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

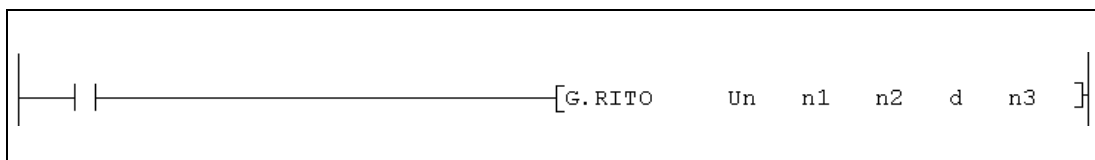
Операнды  
MELSEC Q

	Операнды									Флаг ошибки	Шагов
	Внутренние операнды (системн., пользов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)	Иные		
	битовые	словные		битовые	словные						
n1	●	●	●	—	—	—	—	●	—	SM0	9
n2	●	●	●	—	—	—	—	●	—		
d	—	●	●	—	—	—	—	—	—		
n3	●	●	●	—	—	—	—	●	—		

GX IEC  
Developer



GX  
Developer



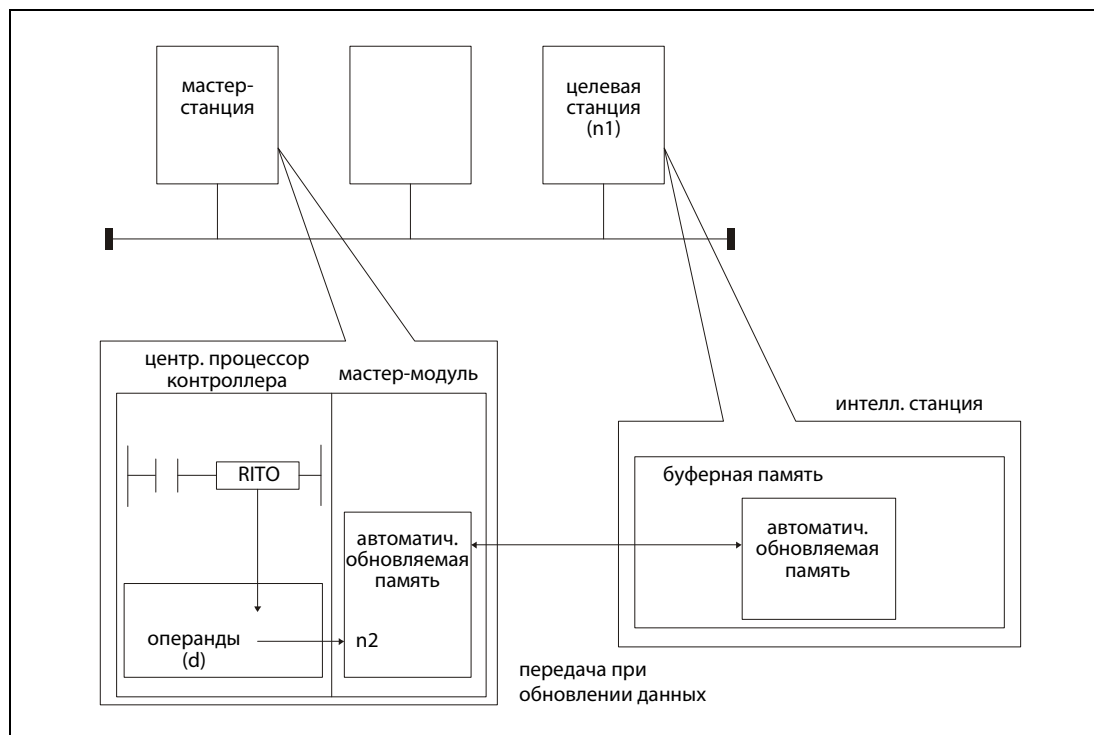
## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных
Un	Головной адрес мастер-модуля CC-Link на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде 10 <sub>H</sub> .)	от 0 до FF <sub>H</sub>	пользователь	BIN, 16 бит
n1	Цель передачи данных <ul style="list-style-type: none"> <li>Укажите номер интеллектуальной станции, в которую требуется передать данные.</li> <li>Если данные требуется передать в буфер со свободным доступом, укажите FF<sub>H</sub>.</li> </ul>	от 1 до 64 или FF <sub>H</sub>		
n2	Величина смещения автоматически обновляемой памяти для интеллектуальной станции в мастер-станции или для буфера со свободным доступом Первый адрес, в который записываются данные, указывается относительно начального адреса области памяти. Пример: чтобы записать данные, начиная с адреса 356 <sub>H</sub> в области памяти, начинающейся с адреса 350 <sub>H</sub> , в n2 необходимо указать значение 6 <sub>H</sub> .	между 0 и макс. значением, установленным в параметр.		
d	Первый операнд области, в которой сохранены передаваемые данные.	действит. адрес операнда		адрес
n3	Количество передаваемых операндов (слов)	от 1 до 4096		BIN, 16 бит

**Принцип действия****Ввод данных в автоматически обновляемую память****RITO      Запись данных**

С помощью команды RITO данные из центрального процессора контроллера передаются в автоматически обновляемую буферную память мастер-станции. Данные определяются их начальным адресом (d) и длиной (n3). Место назначения в мастер-станции указывается в n1 (номер станции, для которой предназначены данные) и n2 (начальный адрес области памяти в мастер-станции). Головной адрес ввода-вывода мастер-станции указан в Un.

Функция команды RITO пояснена следующей иллюстрацией:



К одной и той же интеллектуальной станции невозможно одновременно обращаться из различных станций с помощью нескольких команд RITO.

С помощью команды RITO можно передать максимум 4096 слов.

Присвоить автоматически обновляемые области памяти можно с помощью среды программирования GX Developer или GX IEC Developer в параметрах сетевой коммуникации, пункт "Информация станции".

**Источники ошибок**

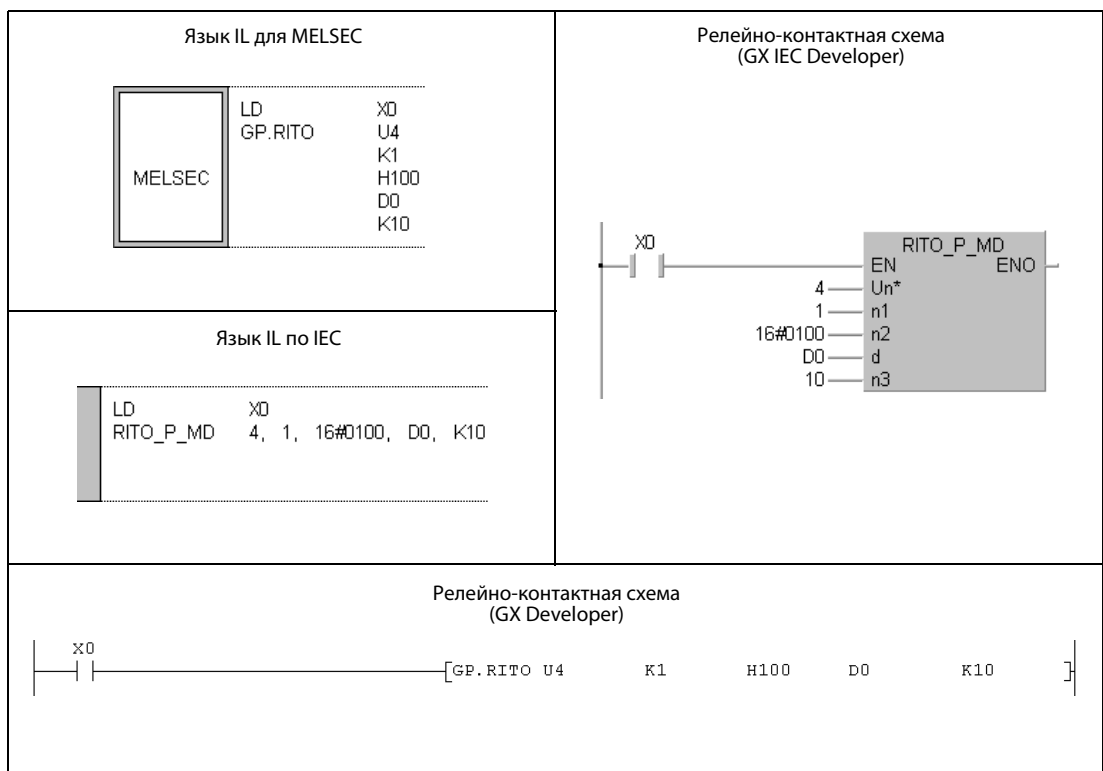
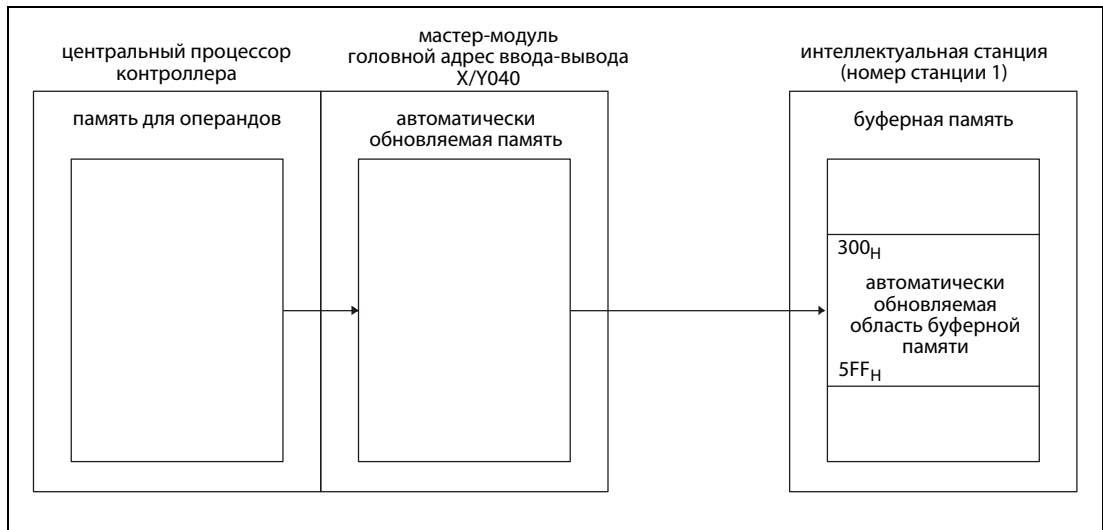
Если при выполнении команды RITO возникла одна из следующих ошибок, то устанавливается флаг ошибки SM0 и в специальный регистр SD0 записывается код ошибки.

- Указанный в Un модуль не является специальным модулем (код ошибки: 2112).
- Выполняемая команда не поддерживается (код ошибки: 4002).
- В команде указано недействительное количество операндов (код ошибки: 4003).
- В команде указан недопустимый операнд (код ошибки: 4004).
- Станция с указанным в n1 номером не существует (код ошибки: 4100).
- Количество данных в n3 превышает допустимый диапазон (код ошибки: 4100).

**Пример**

RITO

При установке входа X0 следующая программа передает содержимое 10 регистров данных D0...D9 в автоматически обновляемую память станции № 1. Эта область памяти начинается с адреса 300<sub>H</sub>. Данные записываются, начиная с адреса 400<sub>H</sub> (смещение = 100).





### 11.5.14 RIFR (серия "A")

**Процессор**

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
●	●	●	●		

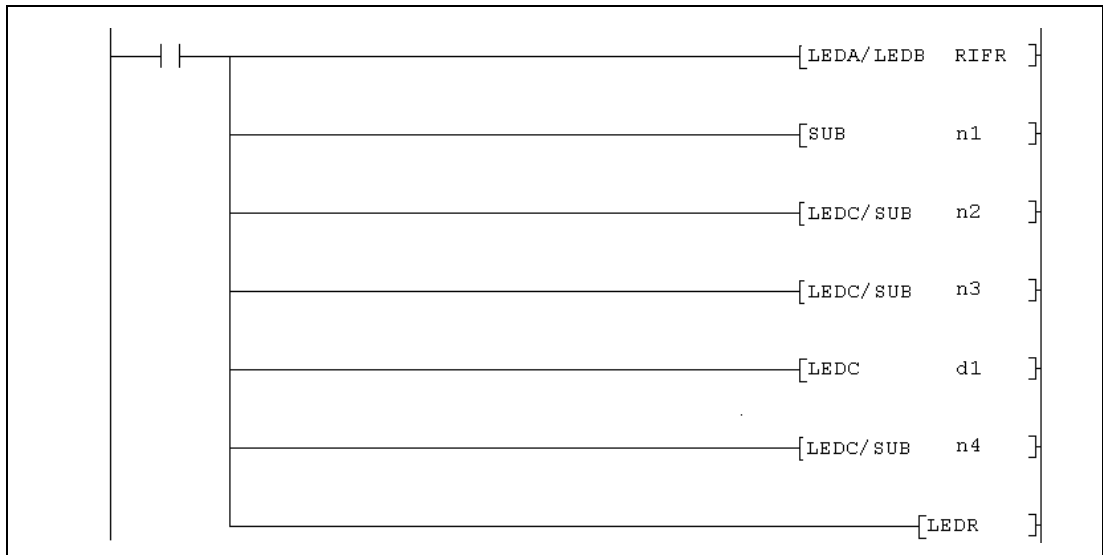
**Операнды MELSEC A**

	Операнды																Длина блока	Шагов	Индекс	Флаг переноса M9012	Флаг ошибки M9011	
	битовые						словные (16 бит)						Константы		Указатели							Уровень
	X	Y	M	L	S	B	F	T	C	D	W	R	A0	A1	Z	V						
n1																	●	●				
n2							●	●	●	●	●					●	●					
n3							●	●	●	●	●					●	●					
n4							●	●	●	●	●					●	●					
d1							●	●	●	●	●											

**GX IEC Developer**

<p style="text-align: center;">Язык IL для MELSEC</p> <table border="1" style="width: 100%;"> <tr> <td style="text-align: center;">MELSEC</td> <td>LEDA/LEDB RIFR</td> </tr> <tr> <td></td> <td>SUB n1</td> </tr> <tr> <td></td> <td>LEDC/SUB n2</td> </tr> <tr> <td></td> <td>LEDC/SUB n3</td> </tr> <tr> <td></td> <td>LEDC d1</td> </tr> <tr> <td></td> <td>LEDC/SUB n4</td> </tr> <tr> <td></td> <td>LEDR</td> </tr> </table>	MELSEC	LEDA/LEDB RIFR		SUB n1		LEDC/SUB n2		LEDC/SUB n3		LEDC d1		LEDC/SUB n4		LEDR	<p style="text-align: center;">Релейно-конт. схема (язык LD)</p>	<p style="text-align: center;">Язык IL по IEC</p> <table border="1" style="width: 100%;"> <tr> <td>RIFR_MD</td> <td>n1, n2, n3, n4, d1</td> </tr> </table>	RIFR_MD	n1, n2, n3, n4, d1
MELSEC	LEDA/LEDB RIFR																	
	SUB n1																	
	LEDC/SUB n2																	
	LEDC/SUB n3																	
	LEDC d1																	
	LEDC/SUB n4																	
	LEDR																	
RIFR_MD	n1, n2, n3, n4, d1																	

**GX Developer**



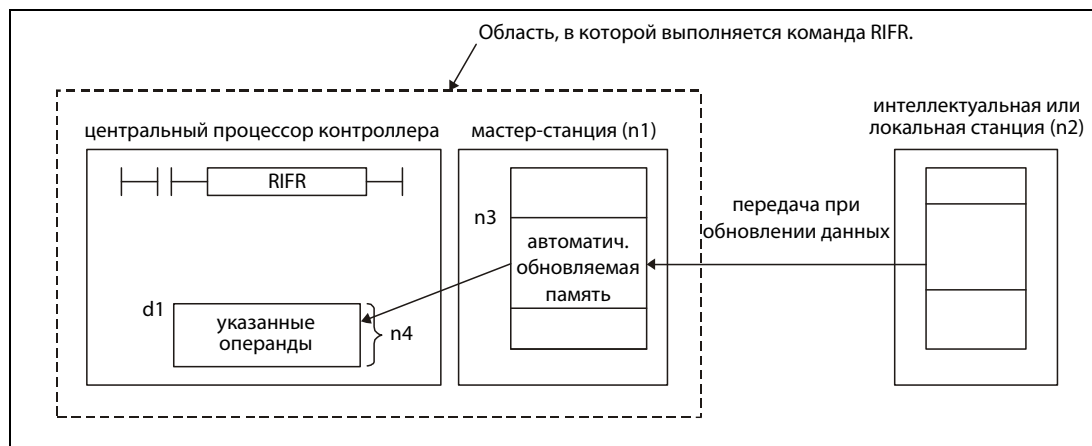
## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных
n1	Головной адрес мастер-модуля CC-Link на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде 10 <sub>H</sub> .)	от 0 до FF <sub>H</sub>	пользователь	BIN, 16 бит
n2	Источник данных Укажите номер интеллектуальной станции, из которой требуется считать данные. (Только если команда RIFR выполняется в мастер-станции). Если данные требуется считать из буфера со свободным доступом, укажите FF <sub>H</sub> .	от 1 до 64 или FF <sub>H</sub>		
n3	Начальный адрес автоматически обновляемой памяти для интеллектуальной станции в мастер-станции или смещение для буфера со свободным доступом	между 0 и макс. значением, установленным в параметр.		
n4	Количество передаваемых операндов (слов)	от 1 до 4096		
d1	Первый операнд области, в которой записываются считанные данные.	действит. адрес операнда		адрес

**Принцип действия****Считывание данных из автоматически обновляемой памяти****RIFR      Считывание данных**

С помощью команды RIFR данные из автоматически обновляемой буферной памяти мастер-станции передаются в память операндов центрального процессора контроллера. В d1 и n4 указывается, где следует сохранить передаваемые данные и какое количество требуется передать. В n2 указывается номер станции, из которой считываются данные, а в n3 – первый считываемый адрес. Головной адрес ввода-вывода мастер-станции указывается в n1.

Функция команды RIFR пояснена следующей иллюстрацией:



С помощью команды RIFR можно передать максимум 4096 слов.

Для установки количества автоматически обновляемых адресов должен быть установлен размер автоматически обновляемого буфера с помощью команды RLPA.

**Условия выполнения**

Если команда RIFR запрограммирована в сочетании с командой LEDA, то команда RIFR выполняется до тех пор, пока условие выполнения команды LEDA включено.

При применении команды LEDB данные передаются только по при положительном фронте условия выполнения.

**Источники ошибок**

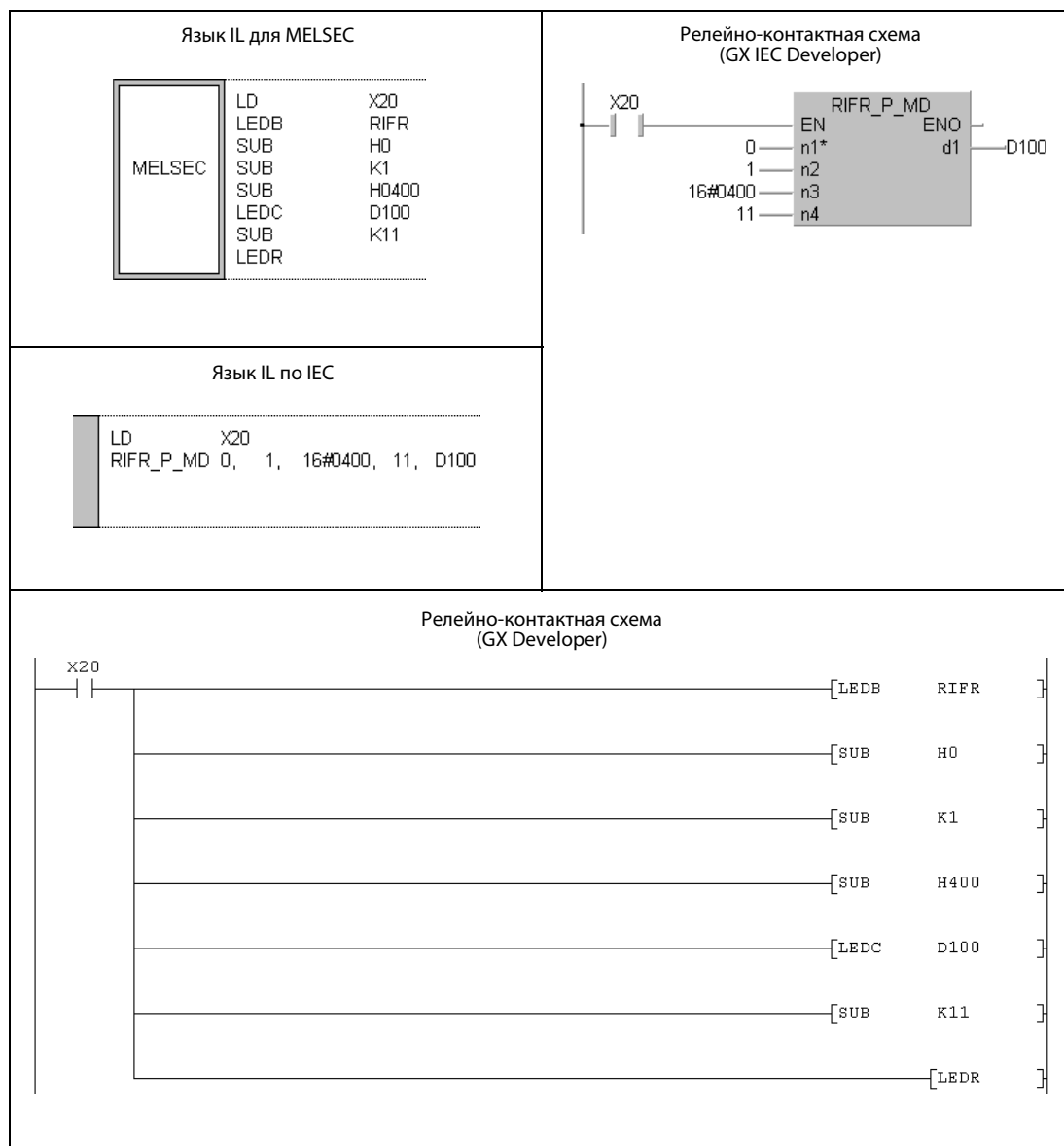
При следующих событиях распознается ошибка обработки, устанавливается флаг ошибки M9011 и выводится код ошибки.

- Настроенный адрес буферной памяти находится вне допустимого диапазона.  
(код ошибки в D9008: 50,  
код ошибки в D9091 (процессоры AnU) или D9092 (процессоры AnSH): 503)
- Количество обновляемых адресов превышает 4096.  
(код ошибки в D9008: 50,  
код ошибки в D9091 (процессоры AnU) или D9092 (процессоры AnSH): 503)

## Пример

## RIFR

При установке входа X20 следующая программа передает содержимое одиннадцати адресов из автоматически обновляемого буфера станции 1 в центральный процессор контроллера и сохраняет в нем данные, начиная с регистра D100. В автоматически обновляемом буфере данные хранятся, начиная с адреса 400<sub>H</sub>. Модуль CC-Link мастер-станции занимает диапазон адресов от X/Y000 до X/Y01F.



Указания по программированию специализированных команд в редакторах MELSEC для контроллеров серии "A" имеются в разделе 3.3 этого руководства.

### 11.5.15 RIFR (серии QnA и "System Q")

Процессор

AnS	AnN	AnA(S)	AnU	QnA(S), Q4AR	System Q
				●	●

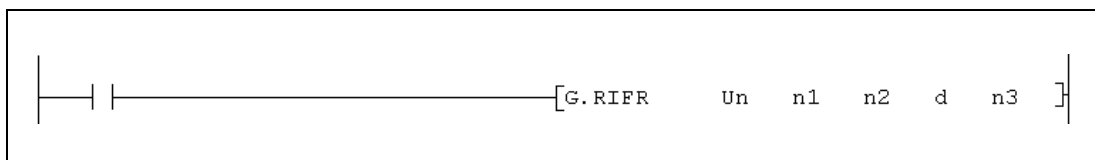
Операнды  
MELSEC Q

	Операнды								Флаг ошибки	Шагов	
	Внутренние операнды (системн., польз.ов.)		Регистры файлов	MELSECNET/10 непоср. адр. J□\□		Спец. модули U□\G□	Индексные регистры Zn	Константы K, H (16#)			Иные
	битовые	словные		битовые	словные						
n1	●	●	●	—	—	—	—	●	—	SM0	9
n2	●	●	●	—	—	—	—	●	—		
n3	●	●	●	—	—	—	—	●	—		
d	—	●	●	—	—	—	—	—	—		

GX IEC  
Developer



GX  
Developer



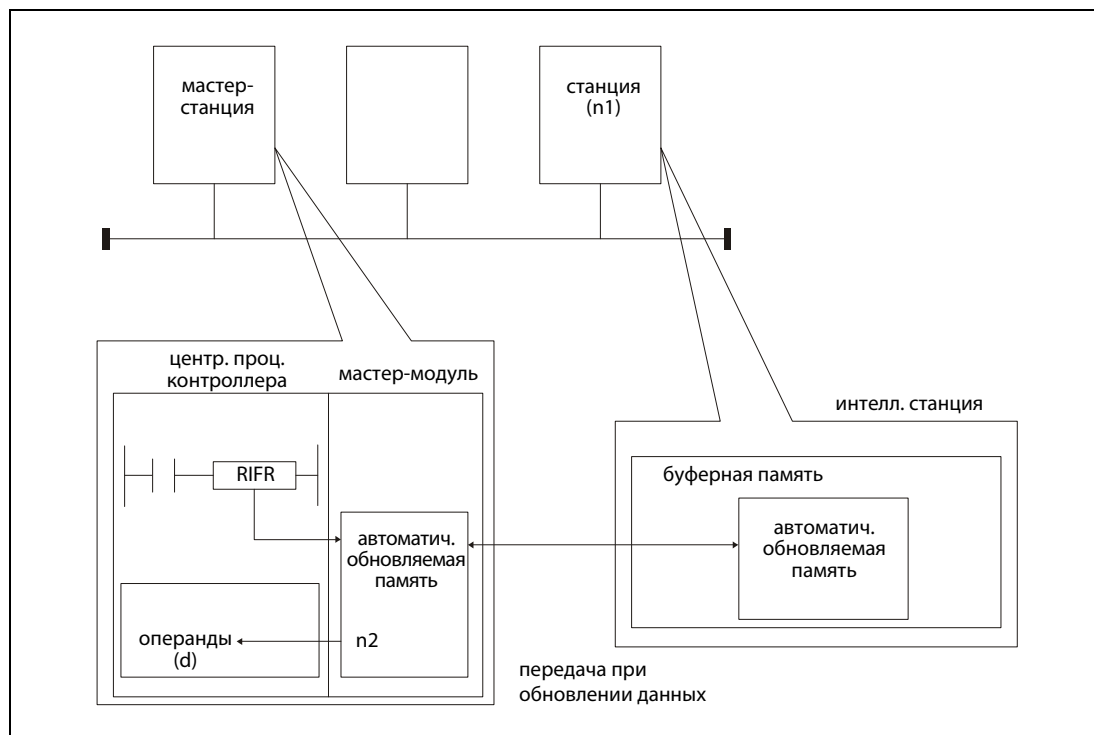
## Переменные

Операнд	Значение	Диапазон	Устанавливает	Тип данных
Un	Головной адрес мастер-модуля CC-Link на монтажной шине (Указываются только первые два разряда 3-разрядного адреса. Например, головной адрес X/Y100 указывается в виде 10 <sub>H</sub> .)	от 0 до FE <sub>H</sub>	пользователь	BIN, 16 бит
n1	Источник данных <ul style="list-style-type: none"> <li>Укажите номер интеллектуальной станции, из которой требуется считать данные.</li> <li>Если данные требуется передать из буфера со свободным доступом, укажите FF<sub>H</sub>.</li> </ul>	от 1 до 64 или FF <sub>H</sub>		
n2	Величина смещения автоматически обновляемой памяти для интеллектуальной станции в мастер-станции или для буфера со свободным доступом Первый считываемый адрес указывается относительно начального адреса области памяти. Пример: чтобы считать данные, начиная с адреса 356 <sub>H</sub> в области памяти, начинающейся с адреса 350 <sub>H</sub> , в n2 необходимо указать значение 6 <sub>H</sub> .	между 0 и макс. значением, установленным в параметр.		
n3	Количество передаваемых операндов (слов)	от 1 до 4096		
d	Первый операнд области, в которой сохраняются считанные данные.	действит. адрес операнда		адрес

**Принцип действия****Считывание данных из автоматически обновляемой памяти****RIFR      Считывание данных**

С помощью команды RIFR данные из автоматически обновляемой буферной памяти мастер-станции передаются в память операндов центрального процессора контроллера. В  $d$  и  $n3$  указывается, где в нем следует сохранить данные и какое количество данных требуется передать. В  $n1$  указывается номер станции, из которой считываются данные, а в  $n2$  (в виде смещения) – первый считываемый адрес. Головной адрес ввода-вывода мастер-станции указывается в  $Un$ .

Функция команды RIFR пояснена следующей иллюстрацией:



К одной и той же интеллектуальной станции невозможно одновременно обращаться из различных станций с помощью нескольких команд RIFR.

С помощью команды RIFR можно передать максимум 4096 слов.

Присвоить автоматически обновляемые области памяти можно с помощью среды программирования GX Developer или GX IEC Developer в параметрах сетевой коммуникации, пункт "Информация станции".

**Источники ошибок**

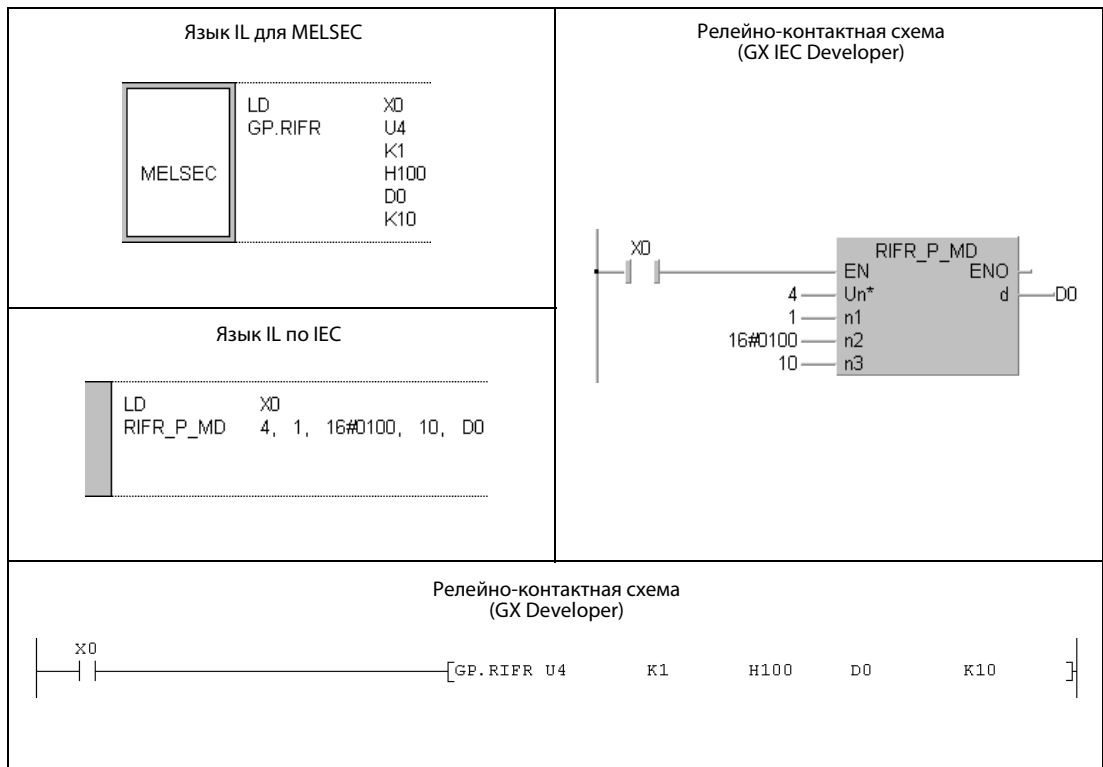
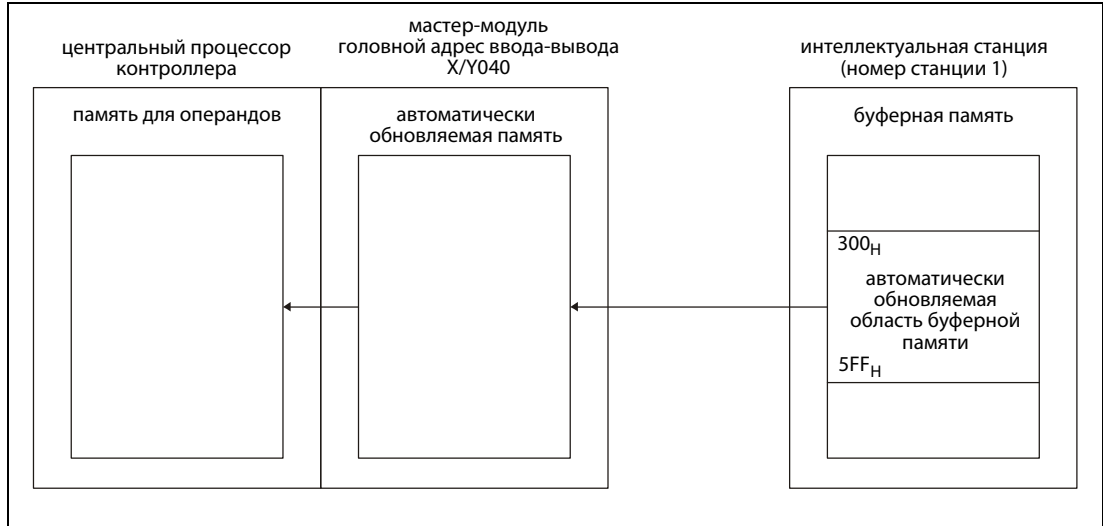
Если при выполнении команды RITO возникла одна из следующих ошибок, то устанавливается флаг ошибки SMO и в специальный регистр SD0 записывается код ошибки.

- Указанный в  $Un$  модуль не является специальным модулем (код ошибки: 2112).
- Выполняемая команда не поддерживается (код ошибки: 4002).
- В команде указано недействительное количество операндов (код ошибки: 4003).
- В команде указан недопустимый операнд (код ошибки: 4004).
- Станция с указанным в  $n1$  номером не существует (код ошибки: 4100).
- Количество данных в  $n3$  превышает допустимый диапазон (код ошибки: 4100).

**Пример**

RIFR

При установке входа X0 следующая программа считывает 10 слов из автоматически обновляемой памяти станции № 1 и сохраняет данные в центральном процессоре по адресу D0. Автоматически обновляемая область памяти начинается с адреса 300<sub>H</sub>. Считываются 10 слов, начиная с адреса 400<sub>H</sub> (смещение = 100).





## 12 Программа микрокомпьютера (AnN(S))

Контроллеры MELSEC серии "A" (кроме AnA, AnAS и AnU) поддерживают комбинированное выполнение основной программы и программ микрокомпьютера. Программа микрокомпьютера позволяет выполнять программные фрагменты вне макроуровня (MAIN и SUB). Программы микрокомпьютера вызываются с помощью команды SUB(P).

Центральные процессоры MELSEC серий AnA, AnAS, AnUS, QnA и QnAS, а также центральные процессоры серии "System Q" не поддерживают программы микрокомпьютера.

### 12.1 Объем и области памяти

В следующей таблице дан обзор возможных размеров программ микрокомпьютера для центральных процессоров различных типов.

ЦПУ	Процессор	Область для программ микрокомпьютера	Рабочая область	Стековая область	Неприменимые команды			
A1	8086 (8 МГц)	0–10 кбайт	A100H–A1FFH (256 байт)	Полезная область: 128 байт	INT, INTO, IRET, IN, OUT, HLT, WAIT, LOCK, ESC			
A2		0–26 кбайт						
A3		0–58 кбайт (MAIN) 0–58 кбайт (SUB)						
A1N	8086 (10 МГц)	0–10 кбайт						
A2N-S1		0–26 кбайт						
A2S		0–26 кбайт						
A3N		0–58 кбайт (MAIN) 0–58 кбайт (SUB)						
A1S	8086 (8 МГц)	0–14 кбайт						
A1S-S1		0–14 кбайт						
A2C		0–26 кбайт						
A3H	80286 (8 МГц)	0–58 кбайт (MAIN) 0–58 кбайт (SUB)						INT, INTO, IRET, IN, OUT, HLT, WAIT, LOCK, ESC, CLI, STI
A3M		0–58 кбайт (MAIN) 0–58 кбайт (SUB)						

#### ПРИМЕЧАНИЯ

Область памяти для программ микрокомпьютера выделяется с шагом в 2 кбайта. Соотношение между отдельными частями программы необходимо выбрать так, чтобы программа микрокомпьютера и основная программа в областях MAIN и SUB не занимали одну и ту же память.

В сочетании с программой микрокомпьютера разрешается использовать только предусмотренные для нее команды. Применение других команд приводит к неправильному функционированию центрального процессора при выполнении программы микрокомпьютера.

## 12.2 Применение самостоятельно созданных программ микрокомпьютера

Исходную программу, созданную пользователем на языке процессора 8086, необходимо компилировать на машинный язык, понятный контроллеру, с помощью ассемблеров в среде CP/M® или MS-DOS®. Компилированная программа называется "конечной программой". Она сохраняется в области для программ микрокомпьютера центрального процессора. Файл типа OBJ передается из компилятора Си в контроллер с помощью программатора.

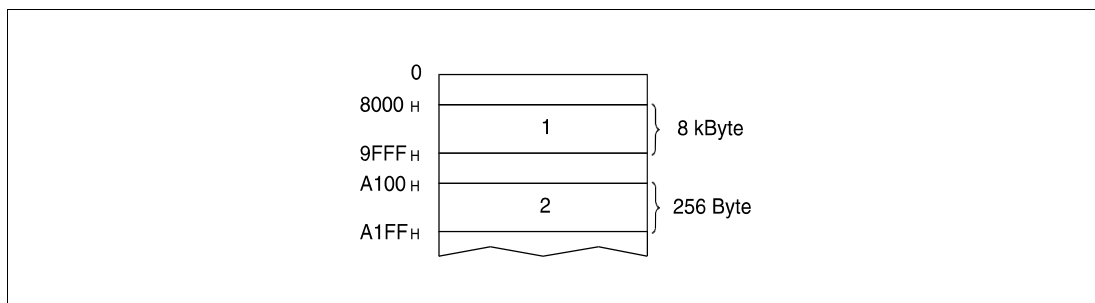
**ПРИМЕЧАНИЕ** Проверьте, поддерживает ли ваша версия программного обеспечения среды GX IEC Developer эти функции.

### Важные указания по созданию программ микрокомпьютера

- В начале программы микрокомпьютера должна находиться команда PUSH, служащая для передачи содержимого используемых регистров данных в стековую память. В конце программы должна находиться команда POP для записи данных, сохраненных в стековой памяти, обратно в регистры данных.
- Все регистры данных, используемые в связи с программой микрокомпьютера, в начале выполнения программы должны инициализироваться, т. е. сбрасываться на исходные значения. После вызова программы микрокомпьютера из основной программы содержимое регистров данных не определено.
- Для выполнения программы микрокомпьютера ее необходимо вызвать из основной программы с помощью команды SUB(P). Поэтому основная программа нужна в любом случае.
- Возврат из программы микрокомпьютера в основную программу осуществляется с помощью команды RETF.

### 12.2.1 Распределение памяти

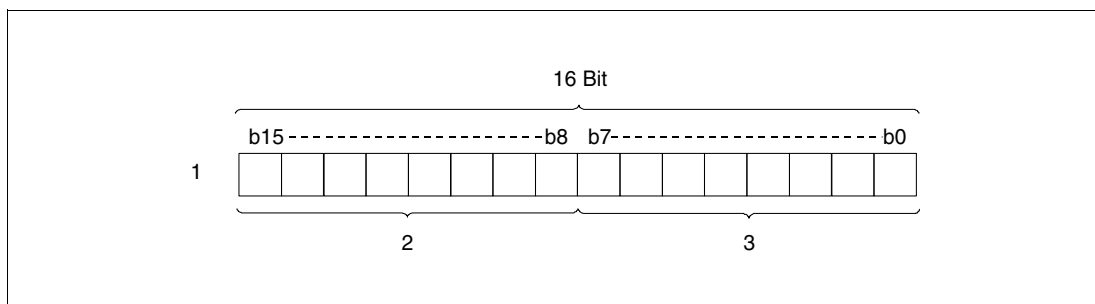
Программа микрокомпьютера сохраняется в двух различных областях памяти центрального процессора. Область размером 8 кбайт, находящаяся между адресами 8000н и 9FFFн, используется для сохранения данных, а область между адресами A100н и A1FFн – в качестве рабочей области программы микрокомпьютера.



- <sup>1</sup> область для сохранения данных
- <sup>2</sup> рабочая область для программа микрокомпьютера

### 12.2.2 Структура адресов в области для сохранения данных

Один адрес области для сохранения данных состоит из 16 битов и подразделяется на четную и нечетную область по 8 битов. На рисунке ниже схематически показана структура адреса.



- <sup>1</sup> первый адрес 8000н
- <sup>2</sup> нечетная область из 8 битов (8001н)
- <sup>3</sup> четная область из 8 битов (8000н)

### 12.2.3 Разбивка области памяти

Для сохранения данных операндов центральный процессор использует область памяти от 8000H до 9FFFH. Разбивка этой области по адресам операндов разъяснена в следующих таблицах.

Операнд	Тип проц.	Адреса	Конфигурация
Вход (X)	A1 A1N	8000H – 803FH	X0–FF
	A2 A2C A2N A2S	8000H – 803FH	X0–1FF
	A2N-S1	8000H – 80FFH	X0–3FF
	A3 A3N	8000H – 81FFH	X0–7FF
Выход (Y)	A1 A1N	8200H – 823FH	Y0–FF
	A2 A2C A2N A2S	8000H – 827FH	Y0–1FF
	A2N-S1	8200H – 827FH	X0–3FF
	A3 A3N	8200H – 83FFH	Y0–7FF

1

2

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
XIM7	XIM6	XIM5	XIM4	XIM3	XIM2	XIM1	XIM0	X7	X6	X5	X4	X3	X2	X1	X0
XIMF	XIME	XIMD	XIMC	XIMB	XIMA	XIM9	XIM8	XF	XE	XD	XC	XB	XA	X9	X8
XIM17	XIM16	XIM15	XIM14	XIM13	XIM12	XIM11	XIM10	X17	X16	X15	X14	X13	X12	X11	X10

3

4

1

2

b15	-----							b8	b7	b6	b5	b4	b3	b2	b1	b0
								Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	
								YF	YE	YD	YC	YB	YA	Y9	Y8	
								Y17	Y16	Y15	Y14	Y13	Y12	Y11	Y10	

5

В режиме отображения ввода-вывода данные записываются в выходной модуль или считываются из выходного модуля через выходную память. При непосредственной обработке ввода-вывода запись происходит непосредственно в выходной модуль. Данные выходного модуля считываются через выходную память.

- 1 нечетные адреса
- 2 четные адреса
- 3 Область для сохранения состояния данных удаленной станции (запись и считывание), 0 = выключено, 1 = включено. Текущий ввод определяется по формуле:  $(X)=(XIM) \vee (X)$ .
- 4 Область для сохранения состояния данных входного модуля (только считывание), 0 = выключено, 1 = включено.
- 5 Область для сохранения результатов обработки в контроллере (возможны считывание и запись).

Операнд	Тип проц.	Адреса	Конфигурация
Маркеры (M) Фиксир. маркеры (L) Шаговые маркеры (S)	A1 A2 A3 A1N A2N-S1 A3N A1S A1S-S1 A2C	8400H – 85FFH	M/L/S 0-2047
Маркеры связи (B)		8600H – 86FFH	F0-255
Маркеры ошибок (F)		8700H – 873FH	F0-255
Спец. маркеры (M)		8740H – 877FH	M 9000-9255
Входной контакт таймера (T)		8780H – 87BFH	T0-255
Входной контакт счетчика (C)		87C0H – 87FFH	C0-255
Выходной контакт таймера (T)		9C00H – 9C3FH	T0-255
Выходной контакт счетчика (C)		9C40H – 9C7FH	C0-255

The diagram shows a memory map with bit fields b15 to b0 and M0 to M17. Brackets 1 and 2 indicate odd and even bit areas. A double arrow 3 points to the M0-M17 fields.

- 1 нечетная область
- 2 четная область
- 3 Область для сохранения результатов обработки в контроллере (возможны считывание и запись).

Операнд	Тип проц.	Адреса	Конфигурация
Регистры данных (D)	A1 A2 A3 A1N A2N-S1 A3N A1S1 A1S-S1 A2C	8800H – 8FFFH	
Регистры связи (W)		9000H – 97FFH	
Фактич. значение таймеров (T)		9800H – 99FFH	
Фактич. значение счетчиков (C)		9A00H – 9BFFH	
Спец. регистры (D)		9D00H – 9EFFH	
Сумматор (A0, A1)		9FF8H – 9FFAH	
Индексные регистры (Z, V)		9FFCH – 9FFEH	

Операнд	Тип проц.	Адреса	Конфигурация	
Вход (X)	АЗН АЗМ	8000H – 80FFH		
Выход (Y)		8200H – 82FFH	<p>В режиме отображения ввода-вывода данные записываются в выходной модуль или считываются из выходного модуля через выходную память. При непосредственной обработке ввода-вывода запись происходит непосредственно в выходной модуль. Данные выходного модуля считываются через выходную память.</p>	
Маркеры (M) Фиксир. маркеры (L) Шаговые маркеры (S)		8400H – 84FFH	M/L/S 0–2047	
Регистры связи (B)		8600H – 867FH	B0–3FF	
Маркеры ошибки (F)		8700H – 871FH	F0–255	

- 1 нечетные адреса
- 2 четные адреса
- 3 Область для сохранения состояния данных входного модуля (только считывание).
- 4 Область для сохранения результатов обработки в контроллере (возможны считывание и запись).

Операнд	Тип проц.	Адреса	Конфигурация																																																																
Спец. маркеры (M)	АЗН АЗМ	8740H – 875FH	<div style="display: flex; justify-content: space-around;"> <span>1</span> <span>2</span> </div> <table border="1" style="border-collapse: collapse; text-align: center; font-size: small;"> <tr> <td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>M15</td><td>M14</td><td>M13</td><td>M12</td><td>M11</td><td>M10</td><td>M9</td><td>M8</td><td>M7</td><td>M6</td><td>M5</td><td>M4</td><td>M3</td><td>M2</td><td>M1</td><td>M0</td> </tr> <tr> <td>M31</td><td>M30</td><td>M29</td><td>M28</td><td>M27</td><td>M26</td><td>M25</td><td>M24</td><td>M23</td><td>M22</td><td>M21</td><td>M20</td><td>M19</td><td>M18</td><td>M17</td><td>M16</td> </tr> <tr> <td>M47</td><td>M46</td><td>M45</td><td>M44</td><td>M43</td><td>M42</td><td>M41</td><td>M40</td><td>M39</td><td>M38</td><td>M37</td><td>M36</td><td>M35</td><td>M34</td><td>M33</td><td>M32</td> </tr> </table> <p style="text-align: center;">↓ 3</p>	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	M15	M14	M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0	M31	M30	M29	M28	M27	M26	M25	M24	M23	M22	M21	M20	M19	M18	M17	M16	M47	M46	M45	M44	M43	M42	M41	M40	M39	M38	M37	M36	M35	M34	M33	M32
b15		b14		b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																		
M15		M14		M13	M12	M11	M10	M9	M8	M7	M6	M5	M4	M3	M2	M1	M0																																																		
M31		M30		M29	M28	M27	M26	M25	M24	M23	M22	M21	M20	M19	M18	M17	M16																																																		
M47		M46		M45	M44	M43	M42	M41	M40	M39	M38	M37	M36	M35	M34	M33	M32																																																		
Входной контакт таймера (T)		8780H – 879FH		T0–255																																																															
Входной контакт счетчика (C)	87C0H – 87DFH	C0–255																																																																	
Выходной контакт таймера (T)	9C00H – 9C1FH	T0–255																																																																	
Выходной контакт счетчика (C)	9C40H – 9C5FH	C0–255																																																																	

- <sup>1</sup> нечетные адреса
- <sup>2</sup> четные адреса
- <sup>3</sup> Область для сохранения результатов обработки в контроллере (возможны считывание и запись).



Операнд	Тип проц.	Адреса	Конфигурация
Регистры данных (D)	АЗН АЗМ	8800H – 8FFFH	
Регистры связи (B)		9000H – 97FFH	
Фактич. значение таймеров (T)		9800H – 99FFH	
Фактич. значение счетчиков (C)		9A00H – 9BFFH	
Спец. регистры (D)		9D00H – 9EFFH	
Сумматор (A0, A1)		9FF8H – 9FFAH	
Индексные регистры (Z, V)		9FFCH – 9FFEH	

Операнд	Тип проц.	Адреса
<p>Регистры файлов (R)</p> <p>№ блока 0</p>	<p>A2</p> <p>A3</p> <p>A2N</p> <p>A2N-S1</p> <p>A2S</p> <p>A3N</p> <p>A3H</p> <p>A3M</p> <p>A1S</p> <p>A1S-S1</p> <p>A2C</p>	<p>Начальный адрес регистров файлов</p> <p>= 20000H + (емкость кассеты RAM) – (объем регистров файлов)</p> <p>Емкость кассет RAM (значения для расчетов)</p> <p>A3(N)MCA – 0 = 16 кбайт A3(N)MCA – 2 = 16 кбайт A3(N)MCA – 4 = 32 кбайт A3(N)MCA – 8 = 64 кбайт A3MCA – 12 = 96 кбайт A3MCA – 16 = 144 кбайт (соотв. полезному объему 128к) A3MCA – 18 = 144 кбайт A3MCA – 24 = 144 кбайт (соотв. полезному объему 192к) A3NMCA – 40 = 144 кбайт (соотв. полезному объему 320к) A3NMCA – 56 = 144 кбайт (соотв. полезному объему 448к)</p> <p>Значение для расчетов</p> <p>Объем памяти для комментариев: (количество комментариев) x 16 байт + 1 кбайт</p> <p>Объем регистров файлов: (количество регистров файлов) x 2 байта</p> <p>Примечание: При расчете памяти следует учитывать, что 1 кбайт равен не 1000 байтам, а 1024 байтам!</p>
<p>Расшир. регистры файлов (R)</p> <p>№ блока 1–9</p>		<p>Начальный адрес регистров файлов для каждого номера блока</p> <p>= 20000H + (емкость кассеты RAM) – (объем для комментариев) – (объем регистров файлов) – (объем для фикс. состояний) – (объем для выбор. трассировки) – 4000H x n</p> <p>Объем для комментариев: (количество комментариев) x 16 байт + 1 кбайт</p> <p>Объем регистров файлов: (количество регистров файлов) x 2 байта</p> <p>Объем для фиксации состояний: сумма установленных байтов</p> <p>Объем для выборочной трассировки: 8 кбайт (если она возможна)</p> <p>n: номер блока</p>

Операнд	Тип проц.	Адреса																																																			
Расшир. регистры файлов (R)  № блока 10–28	A2 A3 A2N A2N-S1 A2S A3N A3H A3M A1S A1S-S1 A2C	Кассеты памяти																																																			
		<table border="1"> <thead> <tr> <th colspan="2" data-bbox="852 465 868 488">1</th> </tr> <tr> <th data-bbox="751 506 767 528">3</th> <th data-bbox="783 506 799 528">4</th> </tr> </thead> <tbody> <tr> <td data-bbox="743 551 759 573">11</td> <td data-bbox="871 551 951 573">38000 н</td> </tr> <tr> <td data-bbox="743 584 759 607">10</td> <td data-bbox="871 584 951 607">3C000 н</td> </tr> <tr> <td> </td> <td> </td> </tr> </tbody> </table>	1		3	4	11	38000 н	10	3C000 н			<table border="1"> <thead> <tr> <th colspan="2" data-bbox="1206 465 1222 488">2</th> </tr> <tr> <th data-bbox="1106 506 1121 528">3</th> <th data-bbox="1137 506 1153 528">4</th> </tr> </thead> <tbody> <tr> <td data-bbox="1098 551 1121 573">28</td> <td data-bbox="1209 551 1289 573">A0000 н</td> </tr> <tr> <td data-bbox="1098 584 1121 607">27</td> <td data-bbox="1209 584 1289 607">A4000 н</td> </tr> <tr> <td data-bbox="1098 618 1121 640">26</td> <td data-bbox="1209 618 1289 640">A8000 н</td> </tr> <tr> <td data-bbox="1098 651 1121 674">25</td> <td data-bbox="1209 651 1289 674">AC000 н</td> </tr> <tr> <td data-bbox="1098 685 1121 707">24</td> <td data-bbox="1209 685 1289 707">B0000 н</td> </tr> <tr> <td data-bbox="1098 719 1121 741">23</td> <td data-bbox="1209 719 1289 741">B4000 н</td> </tr> <tr> <td data-bbox="1098 752 1121 775">22</td> <td data-bbox="1209 752 1289 775">B8000 н</td> </tr> <tr> <td data-bbox="1098 786 1121 808">21</td> <td data-bbox="1209 786 1289 808">BC000 н</td> </tr> <tr> <td data-bbox="1098 819 1121 842">20</td> <td data-bbox="1209 819 1289 842">C0000 н</td> </tr> <tr> <td data-bbox="1098 853 1121 875">19</td> <td data-bbox="1209 853 1289 875">C4000 н</td> </tr> <tr> <td data-bbox="1098 887 1121 909">18</td> <td data-bbox="1209 887 1289 909">C8000 н</td> </tr> <tr> <td data-bbox="1098 920 1121 943">17</td> <td data-bbox="1209 920 1289 943">CC000 н</td> </tr> <tr> <td data-bbox="1098 954 1121 976">16</td> <td data-bbox="1209 954 1289 976">D0000 н</td> </tr> <tr> <td data-bbox="1098 987 1121 1010">15</td> <td data-bbox="1209 987 1289 1010">D4000 н</td> </tr> <tr> <td data-bbox="1098 1021 1121 1043">14</td> <td data-bbox="1209 1021 1289 1043">D8000 н</td> </tr> <tr> <td data-bbox="1098 1055 1121 1077">13</td> <td data-bbox="1209 1055 1289 1077">DC000 н</td> </tr> <tr> <td data-bbox="1098 1088 1121 1111">12</td> <td data-bbox="1209 1088 1289 1111">E4000 н</td> </tr> <tr> <td data-bbox="1098 1122 1121 1144">11</td> <td data-bbox="1209 1122 1289 1144">E8000 н</td> </tr> <tr> <td data-bbox="1098 1155 1121 1178">10</td> <td data-bbox="1209 1155 1289 1178">EC000 н</td> </tr> </tbody> </table>	2		3	4	28	A0000 н	27	A4000 н	26	A8000 н	25	AC000 н	24	B0000 н	23	B4000 н	22	B8000 н	21	BC000 н	20	C0000 н	19	C4000 н	18	C8000 н	17	CC000 н	16	D0000 н	15	D4000 н	14	D8000 н	13	DC000 н	12	E4000 н	11	E8000 н
1																																																					
3	4																																																				
11	38000 н																																																				
10	3C000 н																																																				
2																																																					
3	4																																																				
28	A0000 н																																																				
27	A4000 н																																																				
26	A8000 н																																																				
25	AC000 н																																																				
24	B0000 н																																																				
23	B4000 н																																																				
22	B8000 н																																																				
21	BC000 н																																																				
20	C0000 н																																																				
19	C4000 н																																																				
18	C8000 н																																																				
17	CC000 н																																																				
16	D0000 н																																																				
15	D4000 н																																																				
14	D8000 н																																																				
13	DC000 н																																																				
12	E4000 н																																																				
11	E8000 н																																																				
10	EC000 н																																																				

- <sup>1</sup> A3NMCA-16
- <sup>2</sup> A3NMCA-24, 40 или 56
- <sup>3</sup> № блока
- <sup>4</sup> Начальный адрес



## 13 Коды ошибок

Если во время запуска, при переводе в режим RUN или во время работы программируемого контроллера возникла ошибка, то функция самодиагностики центрального процессора выводит сообщение об ошибке (на светодиодный индикатор или дисплей) и информация об ошибке сохраняется в специальных маркерах (M) или маркерах диагностики (SM) и специальном регистре (D9008) или регистрах диагностики (SD).

## 13.1 Перечень кодов ошибок (процессоры Q00J, Q00 и Q01)

В следующей таблице дан обзор возможных ошибок вместе с сообщениями об ошибках, их вероятными причинами и мерами по устранению. В этой таблице перечислены только сообщения об ошибках, относящиеся к процессорам Q00J, Q00 и Q01.

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики
				RUN	ERROR		
1000	MAIN CPU DOWN	—	—	не горит	мигает/горит	стоп	непрерывно
1010	END NOT EXECUTE	—	—	не горит	мигает	стоп	во время выполнения команды END
1011							
1012							
1101	RAM ERROR	—	—	не горит	мигает	стоп	при включении/при сбросе
1102							
1103							
1104							
1200	OPE. CIRCUIT ERR.	—	—	не горит	мигает	стоп	при включении/при сбросе
1201							
1202							
1300	FUSE BREAK OFF	№ станции/модуля	—	не горит/горит	мигает/горит	стоп/ продолжение <sup>2</sup>	во время выполнения команды END
1310	I/O INT ERROR	№ станции/модуля	—	не горит	мигает	стоп	во время прерывания
1401	SP. UNIT DOWN	№ станции/модуля	—	не горит	мигает	стоп/ продолжение <sup>3</sup>	при включении/при сбросе при обращении к специальному модулю
1402			Локализация ошибки программы				при обращении к специальному модулю
1403			—				во время выполнения команды END

<sup>1</sup> В круглых скобках указаны адреса специальных регистров, в которых сохраняется отдельная информация об ошибках.

<sup>2</sup> Состояние (режим) центрального процессора при возникновении ошибки можно установить с помощью параметров. Состояние светодиодного индикатора изменяется соответственно.

<sup>3</sup> Для каждого модуля в параметрах можно выбрать, должен ли останавливаться центральный процессор при возникновении ошибки в этом модуле.

	Причина	Устранение
	Отключение режима RUN или неисправность в центральном процессоре 1.) Неполадка, вызванная помехами или иными причинами 2.) Аппаратная неисправность	1.) Измерить уровень помех. 2.) Сбросить центральный процессор и переключить его в режим RUN. Если та же ошибка возникает снова, это указывает на аппаратную неисправность. Обратитесь в сервисную службу Mitsubishi.
	Вся программа выполнена без выполнения команды END. 1.) При выполнении команды END эта команда считается в качестве иного кода. 2.) Код команды END изменился на код другой команды.	
	Сбой во внутренней памяти RAM, в которой сохранена основная программа.	Это указывает на аппаратную неисправность центрального процессора. Обратитесь в сервисную службу Mitsubishi.
	Сбой в памяти RAM, используемой в качестве рабочей области центрального процессора.	
	Внутренняя ошибка центрального процессора	
	Ошибка при адресации RAM в центральном процессоре.	
	Неправильно работает электрический контур, отвечающий за обработку индексов в центральном процессоре.	Это указывает на аппаратную неисправность центрального процессора. Обратитесь в сервисную службу Mitsubishi.
	Неправильно работает аппаратура центрального процессора (логика).	
	Неправильно работает электрический контур, отвечающий за обработку основной программы процесса.	
	Неисправен предохранитель выходного модуля.	1.) Проверьте светодиодные индикаторы предохранителей на выходных модулях и замените предохранитель, светодиод которого горит. 2.) Прочтите общую информацию об ошибках на дисплее программатора и замените предохранитель указанного выходного модуля. Можно также на дисплее программатора проверить специальные регистры с SD130 по SD137 и заменить предохранитель того выходного модуля, для которого соответствующий бит установлен на "1".
	Было выполнено прерывание, хотя в системе нет модуля прерываний.	Один из подключенных модулей имеет аппаратную неисправность. Проверьте подключенные модули. Обратитесь в сервисную службу Mitsubishi и опишите неполадки, возникшие в неисправном модуле.
	1.) Доступ к специальному модулю в начале коммуникации не возможен. 2.) Неправильный размер буферной памяти специального модуля.	Аппаратная неисправность центрального процессора. Обратитесь в сервисную службу Mitsubishi.
	Доступ к специальному модулю не возможен.	Это указывает на аппаратную неисправность. Обратитесь в сервисную службу Mitsubishi.
	1.) Во время выполнения команды END не удалось обратиться к специальному модулю. 2.) В специальном модуле обнаружена ошибка.	Запрашиваемый специальный модуль имеет аппаратную неисправность. Обратитесь в сервисную службу Mitsubishi.

## Перечень кодов ошибок процессоров Q00J, Q00 и Q01 (продолжение)

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики
				RUN	ERROR		
1411	CONTROL-BUS ERR.	№ станции/модуля	локализация ошибки программы	не горит	мигает	стоп	при включении/при сбросе
1412							во время выполнения пары команд FROM/TO
1413		—	непрерывно				
1414		—	во время выполнения команды END				
1415		номер монтажной шины					
1500	AC DOWN	—	—	горит	не горит	продолжение	непрерывно
1600	BATTERY ERROR	название дисковогода	—	горит	не горит	продолжение	непрерывно
2000	UNIT VERIFY ERR.	№ станции/модуля	—	не горит/ горит	мигает/горит	стоп/ продолжение <sup>2</sup>	во время выполнения команды END
2100	SP. UNIT LAY ERR.	№ станции/модуля	—	не горит	мигает	стоп	при включении/при сбросе
2103							
2106							
2107							
2110	SP UNIT ERROR	№ станции/модуля	локализация ошибки программы	не горит/ горит	мигает/горит	стоп/ продолжение <sup>2</sup>	во время выполнения команды
2111							STOP → RUN/во время выполнения команды
2112							

<sup>1</sup> В круглых скобках указаны адреса специальных регистров, в которых сохраняется специфическая информация об ошибках.

<sup>2</sup> Состояние (режим) центрального процессора при возникновении ошибки можно установить с помощью параметров. Состояние светодиодного индикатора изменяется соответственно.



Причина	Устранение
<p>После присвоения адресов с помощью параметров доступ к специальному модулю в начале коммуникации не возможен. При возникновении этой ошибки сохраняется адрес ввода-вывода модуля, который он получил при инициализации.</p>	<p>Имеются аппаратные неисправности в специальном модуле, модуле центрального процессора или монтажной шине. Обратитесь в сервисную службу Mitsubishi.</p>
<p>Команды FROM и/или TO не могут быть выполнены из-за ошибки управляющего импульса. При возникновении этой ошибки сохраняется место ее возникновения.</p>	
<p>В шине серии "Q" обнаружена неисправность. Превышено время ожидания.</p>	<p>Возможно, неисправен специальный модуль, центральный процессор или монтажная шина. Обратитесь в сервисную службу Mitsubishi.</p>
<p>В шине серии "Q" обнаружена неисправность.</p>	
<p>В главной или расширительной монтажной шине обнаружена ошибка.</p>	
<p>Кратковременный обрыв электропитания.</p>	<p>Проверьте электропитание.</p>
<p>1.) Напряжение батареи в центральном процессоре снизилось ниже допустимого значения. 2.) Батарея центрального процессора не соединена с центральным процессором.</p>	<p>1.) Замените батарею. 2.) Если батарея служит для внутренней RAM или функции резервирования, подсоедините провод батареи к модулю центрального процессора.</p>
<p>При включении электропитания изменилась информация модуля ввода-вывода. Во время работы модуль ввода-вывода (или специальный модуль) отсоединился от монтажной шины или не был с ней соединен.</p>	<p>Прочтите общую информацию об ошибках на дисплее программатора и проверьте и/или восстановите правильную посадку соответствующих модулей в слоте. Можно также с помощью программатора проверить специальные регистры с SD150 по SD157 и проверить (улучшить) посадку в слотах модулей, бит которых установлен на "1".</p>
<p>Неправильное присвоение адресов в параметрах: Специальному модулю присвоен адрес модуля ввода-вывода (или наоборот). Модулю, не являющемуся модулем центрального процессора, присвоен адрес центрального процессора (или наоборот). Универсальное реле присвоено модулю, который для этого не пригоден.</p>	<p>Установите параметры адресации и согласуйте их с действительной ситуацией.  Сбросьте настройку универсальных реле.</p>
<p>На монтажной шине имеется более одного модуля прерываний QI60.</p>	<p>Установите только один модуль QI60.</p>
<p>1.) В системе установлено более одного модуля MELSECNET/H. 2.) В системе установлено более одного модуля Ethernet. 3.) В системе установлено более двух модулей CC-Link. 4.) В сети MELSECNET/H имеются идентичные номера сетей или станций.</p>	<p>1.) Используйте максимум 1 модуль. 2.) Используйте максимум 1 модуль. 3.) Используйте максимум 2 модуля. 4.) Проверьте номера сетей и станций.</p>
<p>Головной адрес, присвоенный для сопоставления адресов в параметрах, совпадает с адресом другого модуля.</p>	<p>Сбросьте параметры адресации и согласуйте их с действительной ситуацией.</p>
<p>Модуль, к которому обращается команда FROM/TO, не является специальным модулем. Запрашиваемый специальный модуль имеет неполадку.</p>	<p>Прочтите специфическую информацию об ошибках и проверьте (измените) программирование команд FROM и TO. Если специальный модуль неисправен, обратитесь в сервисную службу Mitsubishi.</p>
<p>Модуль, к которому программа обращается с помощью непосредственно адресуемых операндов связи, не является модулем сетевой коммуникации.</p>	
<p>Запрашиваемый специальный модуль не является специальным модулем или является неправильным специальным модулем.</p>	<p>Прочтите специфическую информацию об ошибках и проверьте (измените) программирование команды.</p>

## Перечень кодов ошибок процессоров Q00J, Q00 и Q01 (продолжение)

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики
				RUN	ERROR		
2120	SP. UNIT LAY ERR.	—	—	не горит	мигает	стоп	при включении/при сбросе
2122							
2124							
2125							
2200	MISSING PARA.	название дисковогода	—	не горит	мигает	стоп	при включении/при сбросе
2400	FILE SET ERROR	название файла	номер параметра	не горит	мигает	стоп	при включении/при сбросе
2401							
2500	CAN'T EXE. PRG.	название файла	—	не горит	мигает	стоп	при включении/при сбросе
2501							
2502							
2503							
3000	PARAMETER ERROR	название файла	номер параметра	не горит	мигает	стоп	при включении/сбросе/переходе STOP → RUN
3001							
3003							
3004							

<sup>1</sup> В круглых скобках указаны адреса специальных регистров, в которых сохраняется специфическая информация об ошибках.

<sup>2</sup> Состояние (режим) центрального процессора при возникновении ошибки можно установить с помощью параметров. Состояние светодиодного индикатора изменяется соответственно.

Причина	Устранение
Установлена монтажная шина QA[]B или QA1S[]B.	Используйте монтажную шину Q[]B
В качестве главной монтажной шины установлена шина QA1S[]B.	В качестве главной монтажной шины используйте Q[]B.
<p>Один из модулей вставлен в 25-й слот или слот с более высоким номером (в случае процессора Q00J – 17-й или более высокий слот).</p> <p>Модуль вставлен в слот, который более не предусмотрен в диапазоне адресов.</p> <p>Модуль занимает адреса входов/выходов, находящиеся вне допустимого диапазона адресов ввода-вывода.</p> <p>Модуль, занимающий последний адрес ввода-вывода, занимает и другие адреса.</p> <p>Подключено более 4 расширительных монтажных шин (в случае процессора Q00J – более 2 расширительных монтажных шин).</p>	<p>Удалите все модули, установленные начиная с 65-го слота.</p> <p>Удалите модуль, вставленный в слот вне выделенного диапазона.</p> <p>Удалите модуль, который занимает адреса ввода-вывода, находящиеся вне допустимого диапазона.</p> <p>Замените этот модуль модулем, не превышающим последний адрес ввода-вывода.</p> <p>Подключите максимум 4 или 2 расширительных монтажных шины.</p>
Установлен модуль, не распознаваемый центральным процессором Q. Доступ к специальному модулю не возможен.	Используйте модуль, совместимый с центральным процессором Q. Запрашиваемый специальный модуль имеет аппаратную неисправность. Обратитесь в сервисную службу Mitsubishi.
В памяти для хранения программ нет никаких параметров.	Проверьте, правильно ли указаны дисководы в настройках параметров. Сохраните параметры в правильном дисковом.
Не найден файл, указанный в параметрах в файловых настройках контроллера.	Прочтите специфическую информацию об ошибках на дисплее программатора и проверьте, соответствуют ли названия дисковода и файла настройкам параметров. Если необходимо, откорректируйте. Создайте заданный файл.
В области протокола ошибок не удалось создать файл, указанный в параметрах в настройках RAS контроллера.	Прочтите специфическую информацию об ошибках на дисплее программатора и проверьте соответствие программы настройкам параметров. Если необходимо, откорректируйте настройки.
Имеется программный файл, который использует операнды вне диапазона, установленного в параметрах операндов.	Прочтите специфическую информацию об ошибках на дисплее программатора и убедитесь в том, что настройки операндов параметров и операндов программного файла соответствуют действительной ситуации. Если необходимо, откорректируйте.
Имеются программные файлы, хотя в параметрах указано "Не имеется".	Измените программные параметры на "Имеются". Сотрите ненужные программы.
Ошибочный программный файл. Содержимое файлов написано не на языке SFC.	Проверьте, имеет ли файл формат *.PGM и предназначено ли содержимое файла для основной программы.
Программный файл не существует.	Проверьте конфигурацию программы.
<p>Настройки параметров для установки таймеров, контакта RUN-PAUSE, общего адреса указателей, обработки всех данных, количества свободных слотов или настройки системных прерываний находятся вне диапазона, который может использовать центральный процессор.</p> <p>Содержимое параметров уничтожено.</p> <p>Установленное в параметрах операндов количество операндов находится вне диапазона, который может использовать центральный процессор.</p>	<p>1.) Прочтите подробную информацию об ошибках на дисплее программатора. Проверьте настройки параметров и, если необходимо, сделайте коррекции.</p> <p>2.) Если ошибка продолжает иметь место, то, вероятно, имеется сбой внутренней памяти RAM центрального процессора.</p>
Файл параметров не может использоваться в процессоре QnA или не содержит параметры.	

## Перечень кодов ошибок процессоров Q00J, Q00 и Q01 (продолжение)

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики
				RUN	ERROR		
3100	LINK PARA. ERROR	название файла	номер параметра	не горит	мигает	стоп	при включении/сбросе/переходе STOP → RUN
3101							
3102							
3103							
3104							
3105							
3106							
3107							
3300	SP. PARA. ERROR	название файла	номер параметра	не горит	мигает	стоп	при включении/сбросе/переходе STOP → RUN
3301							
3302							
4000	INSTRCT CODE. ERR.	локализация ошибки программы	—	не горит	мигает	стоп	при включении/сбросе/переходе STOP → RUN
4002							
4003							
4004							
4010	MISSING END INS.	локализация ошибки программы	—	не горит	мигает	стоп	при включении/сбросе/переходе STOP → RUN
4021	CAN'T SET (P)						
4030	CAN'T SET (I)						

<sup>1</sup> В круглых скобках указаны адреса специальных регистров, в которых сохраняется специфическая информация об ошибках.

Причина	Устранение
<p>Количество установленных модулей отличается от количества, указанного в параметрах для MELSECNET/H.</p> <p>Начальный адрес установленных модулей отличается от начального адреса, указанного в параметрах для MELSECNET10H.</p> <p>Не все данные в параметрах могут быть считаны.</p> <p>В сети MELSECNET/H при включенном напряжении был изменен тип станции. (Чтобы система распознала измененный тип, ее необходимо перевести из состояния RESET в состояние RUN).</p>	<p>1.) После коррекции запишите сетевые параметры заново.</p> <p>2.) Если ошибка продолжает иметь место и после исправления, обратитесь в сервисную службу Mitsubishi.</p>
<p>Номер коммуникационной сети, заданный в параметре, не совпадает с номером установленной сети.</p> <p>Начальный адрес установленных модулей ввода-вывода отличается от начального адреса, заданного в параметрах.</p> <p>Класс коммуникационной сети, заданный в параметре, не совпадает с классом действительно установленной сети.</p> <p>Имеется ошибка в параметрах обновления MELSECNET/H.</p>	<p>Приведите параметры в соответствии с установленной коммуникационной сетью.</p>
<p>При проверке параметров сети модуля сетевой коммуникации возникла ошибка.</p>	<p>1.) После коррекции запишите сетевые параметры заново.</p> <p>2.) Если ошибка продолжает иметь место и после исправления, обратитесь в сервисную службу Mitsubishi.</p>
<p>Хотя в параметрах задан как минимум один модуль Ethernet, ни один модуль не установлен.</p> <p>Начальный адрес установленных модулей ввода-вывода отличается от начального адреса, заданного в параметрах Ethernet.</p>	
<p>Сетям Ethernet и MELSECNET/10 присвоены одни и те же номера сетей.</p> <p>Данные в параметрах номера сети, номера станции или номера группы превышают допустимый диапазон.</p> <p>Заданный адрес ввода-вывода превышает диапазон, допустимый для центрального процессора.</p> <p>Ошибка параметра, относящегося к Ethernet.</p>	
<p>Хотя в параметрах указан как минимум один модуль CC-Link, ни один модуль не установлен.</p> <p>Начальный адрес области ввода-вывода, указанный в общих параметрах, отличается от установленного модуля ввода-вывода.</p> <p>Класс станции CC-Link, заданный в параметре, не совпадает с классом действительно установленной станции.</p>	<p>1.) После коррекции запишите сетевые параметры заново.</p> <p>2.) Если ошибка продолжает иметь место и после исправления, обратитесь в сервисную службу Mitsubishi.</p>
<p>Имеется ошибка в параметрах обновления CC-Link.</p>	<p>Проверьте параметрирование.</p>
<p>Ошибочное содержимое параметров CC-Link.</p>	
<p>Головной адрес для специального модуля, присвоенный средой GX Configurator, не совпадает с действительным адресом ввода-вывода.</p>	
<p>При обновлении специального модуля превышает диапазон для регистра файлов.</p> <p>Ошибка в параметрах обновления специального модуля.</p>	<p>Проверьте параметрирование.</p>
<p>Ошибочное параметрирование специального модуля.</p>	
<p>В программе имеется код команды, который не удалось декодировать</p> <p>Программа содержит команду, которая не может быть выполнена.</p>	
<p>Команда имеет неправильное название.</p>	<p>Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.</p>
<p>Команда обращается к неправильному адресу операнда.</p>	
<p>Команда обращается к операнду, который нельзя использовать.</p>	
<p>В программе нет команды END (FEND)</p>	
<p>Перекрываются адреса общих указателей, используемые соответствующими файлами.</p>	<p>Перекрываются адреса присвоенных указателей, используемые соответствующими файлами.</p>
<p>Перекрываются адреса присвоенных указателей, используемые соответствующими файлами.</p>	

## Перечень кодов ошибок процессоров Q00J, Q00 и Q01 (продолжение)

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики
				RUN	ERROR		
4100	OPERATION ERROR	локализация ошибки программы	—	не горит/горит	мигает/горит	стоп/продолжение <sup>2</sup>	при выполнении команды
4101							
4102							
4108							
4200	FOR NEXT ERROR	локализация ошибки программы	—	не горит	мигает	стоп	при выполнении команды
4201							
4202							
4203							
4210	CAN'T EXECUTE ( P )	локализация ошибки программы	—	не горит	мигает	стоп	при выполнении команды
4211							
4212							
4213							
4220	CAN'T EXECUTE ( I )	локализация ошибки программы	—	не горит	мигает	стоп	при выполнении команды
4221							
4223							
4231	INST. FORMAT ERR	локализация ошибки программы	—	не горит	мигает	стоп	при выполнении команды
5001	WDT ERROR	время (настройка)	время (фактическое измеренное значение)	не горит	мигает	стоп	непрерывно
5010	PRG. TIME OVER	время (настройка)	время (фактическое измеренное значение)	горит	горит	продолжение	непрерывно
9000	r**** <sup>3</sup>	локализация ошибки программы	№ маркера ошибки	горит	не горит	продолжение	при выполнении команды
				горит светодиод "USER"			

<sup>1</sup> В круглых скобках указаны адреса специальных регистров, в которых сохраняется специфическая информация об ошибках.

<sup>2</sup> Состояние (режим) центрального процессора при возникновении ошибки можно установить с помощью параметров. Состояние светодиодного индикатора изменяется соответственно.

<sup>3</sup> \*\*\*\* указывают на распознанный номер маркера ошибки.

Причина	Устранение
<p>Содержащиеся данные не могут обрабатываться соответствующей командой.</p> <p>Указанные адреса данных, которые должна обрабатывать программа, или сохраненные данные или константы операндов, используемых командами, находятся вне диапазона адресов, который можно использовать.</p> <p>Неправильный номер сети или станции, к которому обращается специализированная команда сетевой коммуникации.</p>	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.
При выполнении команды CC-Link не были введены параметры CC-Link.	Перед выполнением команды CC-Link установите параметры.
<p>После команды FOR не выполняется команда NEXT, или имеется меньшее количество команд NEXT, чем команд FOR.</p> <p>Выполняется команда NEXT, хотя не была выполнена команда FOR, или имеется больше команд NEXT, чем команд FOR.</p>	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (исправьте) указанный шаг программы.
Запрограммировано более 16 уровней вложения.	Уменьшите количество уровней вложения, так чтобы их было менее 17.
Выполняется команда BREAK, хотя не была выполнена команда FOR.	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.
<p>Выполняется команда CALL, однако на используемом указателе нет подпрограммы.</p> <p>В выполненной подпрограмме нет команды RET.</p> <p>Команда RET находится перед командой FEND в основной программе.</p>	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.
<p>Запрограммировано более 16 уровней вложения.</p> <p>Сделан ввод прерывания, однако соответствующий указатель прерывания не обнаружен.</p> <p>В выполненной программе прерывания нет команды IRET.</p> <p>Команда IRET в основной программе находится перед командой FEND.</p>	Уменьшите количество уровней вложения, так чтобы их было менее 17.
Команды IX и IXEND не запрограммированы совместно. Имеется одинаковое количество команд IX и IXEND.	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (исправьте) указанный шаг программы.
Время цикла программы превышает контрольное время сторожевых таймеров, настроенное в параметрах RAS контроллера.	Прочтите специфическую информацию об ошибках на дисплее программатора и проверьте (уменьшите) настроенное время цикла.
Время цикла программы в замедленном режиме обработки ("Low speed scan") превышает постоянное время цикла, настроенное в параметре RAS контроллера.	Проверьте и измените постоянное время цикла или время выполнения программы в замедленном режиме ("low speed scan").
Маркер ошибки F установлен в состояние "включен".	Прочтите специфическую информацию об ошибках на дисплее программатора и проверьте (исправьте) программу на основе номера установленного маркера ошибки.

## 13.2 Перечень кодов ошибок (серии QnA и "System Q")

В следующей таблице дан обзор возможных ошибок вместе с сообщениями об ошибках, их вероятными причинами и мерами по устранению. В этой таблице перечислены лишь сообщения об ошибках, действительные для центральных процессоров Q02(H), Q06H, Q12(P)H, Q25(P)H, QnA(S) и Q4AR. Знак "●" в последнем столбце означает, что данный код ошибки относится ко всем указанным центральным процессорам. "Rem" означает совместимость с удаленными модулями ввода-вывода. Если в этом столбце указан тип центрального процессора, то это означает, что данный код ошибки относится только к центральному процессору данного типа.

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики
				RUN	ERROR		
1000	MAIN CPU DOWN	—	—	не горит	мигает/горит	стоп	непрерывно
1001							
1002							
1003							
1004							
1005							
1006							
1007							
1008							
1009							
1010	END NOT EXECUTE	—	—	не горит	мигает	стоп	во время выполнения команды END
1011							
1012							
1101	RAM ERROR	—	—	не горит	мигает	стоп	при включении/при сбросе
1102							
1103							
1104							
1105							
1200	OPE. CIRCUIT ERR.	—	—	не горит	мигает	стоп	при включении/при сбросе
1201							
1202							
1203							
1204							
1205							
1206							
							во время выполнения команды END
							при выполнении команды

<sup>1</sup>В круглых скобках указаны адреса специальных регистров, в которых сохраняется отдельная информация об ошибках.



## Перечень кодов ошибок (серии QnA и "System Q")

	Причина	Устранение	Отно-ситсяк:
	Отключение режима RUN или неисправность в центральном процессоре 1.) Неполадка, вызванная помехами или иными причинами 2.) Аппаратная неисправность	1.) Измерить уровень помех. 2.) Сбросить центральный процессор и переключить его в режим RUN. Если та же ошибка возникает снова, это указывает на аппаратную неисправность. Обратитесь в сервисную службу Mitsubishi.	QnA
	Отключение режима RUN или неисправность в центральном процессоре 1.) Неполадка, вызванная помехами или иными причинами. 2.) Аппаратная неисправность	1.) Измерить уровень помех. 2.) Сбросить центральный процессор и переключить его в режим RUN. Если та же ошибка возникает снова, это указывает на аппаратную неисправность. Обратитесь в сервисную службу Mitsubishi.	Q
			Q/Rem
			Q
	Вся программа выполнена без выполнения команды END. 1.) При выполнении команды END эта команда считывается в качестве иного кода. 2.) Код команды END изменился на код другой команды.	1.) Измерить уровень помех. 2.) Сбросить центральный процессор и переключить его в режим RUN. Если та же ошибка возникает снова, это указывает на аппаратную неисправность. Обратитесь в сервисную службу Mitsubishi.	
	Сбой во внутренней памяти RAM, в которой сохранена основная программа.	Это указывает на аппаратную неисправность центрального процессора. Обратитесь в сервисную службу Mitsubishi.	●
	Сбой в памяти RAM, используемой в качестве рабочей области центрального процессора.		
	Внутренняя ошибка центрального процессора		
	Ошибка при адресации RAM в центральном процессоре		
	Сбой в общей области памяти для многопроцессорного режима	1.) Измерить уровень помех. 2.) Сбросить центральный процессор и переключить его в режим RUN. Если та же ошибка возникает снова, это указывает на аппаратную неисправность. Обратитесь в сервисную службу Mitsubishi.	Q, начиная с версии "B"
	Неправильно работает электрический контур центрального процессора, отвечающий за обработку индексов.	Это указывает на аппаратную неисправность центрального процессора. Обратитесь в сервисную службу Mitsubishi.	●
	Неправильно работает аппаратура центрального процессора (логика).		
	Неправильно работает электрический контур центрального процессора, отвечающий за обработку основной программы.		
	Неправильно работает электрический контур центрального процессора, отвечающий за обработку индексов.	Это указывает на аппаратную неисправность центрального процессора. Обратитесь в сервисную службу Mitsubishi.	Q4AR
	Неправильно работает аппаратура центрального процессора (логика).		
	Неправильно работает электрический контур, отвечающий за обработку основной программы процесса.		
	Неправильно работает электрический контур центрального процессора, отвечающий за обработку цифровых сигналов.		

## Перечень кодов ошибок (серии QnA и "System Q")

### Перечень кодов ошибок серий QnA и "System Q" (продолжение)

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики
				RUN	ERROR		
1300	FUSE BREAK OFF	№ станции/модуля	—	не горит/ горит	мигает/горит	стоп/ продолжение <sup>2</sup>	во время выполнения команды END
1301	EX POWER OFF	№ станции/модуля	—	не горит/ горит	мигает/горит	стоп/ продолжение <sup>2</sup>	во время выполнения команды END
1310	I/O INT ERROR	№ станции/модуля	—	не горит	мигает/горит	стоп	во время прерывания
1401	SP. UNIT DOWN	№ станции/модуля	—	не горит/ горит	мигает/горит	стоп/ продолжение <sup>3</sup>	при включении/при сбросе при обращении к специальному модулю
1402			локализация ошибки программы				при обращении к специальному модулю
1403			—				во время выполнения команды END
1411	CONTROL-BUS ERR.	№ станции/модуля	локализация ошибки программы	не горит	мигает	стоп	при включении/при сбросе
1412							во время выполнения пары команд FROM/TO
1413	CONTROL-BUS ERR.	—	—	не горит	мигает	стоп	непрерывно

<sup>1</sup> В круглых скобках указаны адреса специальных регистров, в которых сохраняется специфическая информация об ошибках.

<sup>2</sup> Состояние (режим) центрального процессора при возникновении ошибки можно установить с помощью параметров. Состояние светодиодного индикатора изменяется соответственно.

<sup>3</sup> Это сообщение об ошибке возможно только в избыточной системе. Ошибка может распознаваться для активной и резервной системы.

<sup>4</sup> В параметрах можно для каждого модуля установить, как должна реагировать система при наличии ошибки.

## Перечень кодов ошибок (серии QnA и "System Q")

	Причина	Устранение	Отно-сится к:
	Неисправен предохранитель выходного модуля.	1.) Проверьте светодиод "ERR" на модулях вывода и замените модуль, светодиод которого горит. 2.) Модуль с неисправным предохранителем можно найти и путем просмотра специальных регистров с SD1300 по SD1331.	Q Rem
	Неисправен предохранитель выходного модуля.	1.) Проверьте светодиодные индикаторы предохранителей на выходных модулях и замените предохранитель, светодиод которого горит. 2.) Прочтите общую информацию об ошибках на дисплее программатора и замените предохранитель указанного выходного модуля. Другой способ – проверьте специальные регистры с SD1300 по SD1331 с помощью программатора и замените предохранитель выходного модуля, бит которого установлен на "1".	QnA, Q4AR
	Неисправен предохранитель выходного модуля. Внешнее электропитание выходов отключено или не подсоединено.	1.) Проверьте светодиодные индикаторы предохранителей на выходных модулях и замените предохранитель, светодиод которого горит. 2.) Прочтите общую информацию об ошибках на дисплее программатора и замените предохранитель указанного выходного модуля. Другой способ – проверьте специальные регистры с SD1300 по SD1331 с помощью программатора и замените предохранитель выходного модуля, бит которого установлен на "1". 3.) Проверьте внешнее напряжение питания выходов.	Q2AS
	Внешнее электропитание выходов отключено или не подсоединено (это сообщение предусмотрено для использования в будущем).	Проверьте внешнее напряжение питания выходов.	Q/Rem
	Было выполнено прерывание, хотя в системе нет модуля прерываний.	Один из подключенных модулей имеет аппаратную неисправность. Проверьте подключенные модули. Обратитесь в сервисную службу Mitsubishi.	●
	1.) Доступ к специальному модулю в начале коммуникации не возможен. 2.) Неправильный размер буферной памяти специального модуля.	Аппаратная неисправность центрального процессора. Обратитесь в сервисную службу Mitsubishi.	Q Rem
	После присвоения адресов с помощью параметров доступ к специальному модулю в начале коммуникации не возможен. При возникновении этой ошибки сохраняется адрес ввода-вывода модуля, который он получил при инициализации.	Специальный модуль, к которому осуществлялся доступ, имеет аппаратную неисправность. Обратитесь в сервисную службу Mitsubishi.	QnA
	Доступ к специальному модулю не возможен.	Аппаратная неисправность центрального процессора. Обратитесь в сервисную службу Mitsubishi.	Q Rem
	Обращение к специальному модулю с помощью команды FROM и/или TO не сопровождается реакцией специального модуля. При возникновении этой ошибки сохраняется место ее возникновения.	Запрашиваемый специальный модуль имеет аппаратную неисправность. Обратитесь в сервисную службу Mitsubishi.	QnA
	1.) Во время выполнения команды END не удалось обратиться к специальному модулю. 2.) В специальном модуле обнаружена ошибка.		Q Rem
	После присвоения адресов с помощью параметров доступ к специальному модулю в начале коммуникации не возможен. При возникновении этой ошибки сохраняется адрес ввода-вывода модуля, который он получил при инициализации.	Имеются аппаратные неисправности в специальном модуле, модуле центрального процессора или монтажной шине. Обратитесь в сервисную службу Mitsubishi.	● Rem
	Команды FROM и/или TO не могут быть выполнены из-за ошибки управляющего импульса. При возникновении этой ошибки сохраняется место ее возникновения.		●
	В многопроцессорной системе установлен центральный процессор версии "А".	Замените центральный процессор на процессор версии "В" или выше. Возможно, неисправен специальный модуль, центральный процессор или монтажная шина. Обратитесь в сервисную службу Mitsubishi.	Q, начиная с версии "В"
	В шине серии "Q" обнаружена неисправность. Превышено время ожидания.	Возможно, неисправен специальный модуль, центральный процессор или главная монтажная шина. Обратитесь в сервисную службу Mitsubishi.	Q Rem

## Перечень кодов ошибок (серии QnA и "System Q")

### Перечень кодов ошибок серий QnA и "System Q" (продолжение)

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики
				RUN	ERROR		
1414	CONTROL-BUS ERR.	№ станции/модуля	—	не горит	мигает	стоп	во время выполнения команды END
1415		номер монтажной шины					
1416		№ станции/модуля					
1421	SYS. UNIT DOWN <sup>3</sup>	—	—	не горит	мигает	стоп	непрерывно
1500	AC DOWN	—	—	горит	не горит	продолжение	непрерывно
1510	DUAL DC DOWN 5V <sup>4</sup>	—	—	горит	горит	продолжение	непрерывно
1520	DC DOWN 5V <sup>5</sup>	—	—	не горит	мигает	стоп	непрерывно
1530	DC DOWN 24V <sup>3</sup>	—	—	горит	горит	продолжение	непрерывно
1600	BATTERY ERROR	название дисковогода	—	горит	не горит	продолжение	непрерывно
1601				горит светодиод "BAT. ALM"			
1602							
2000	UNIT VERIFY ERR.	№ станции/модуля	—	не горит/ горит	мигает/горит	стоп/ продолжение <sup>2</sup>	во время выполнения команды END
2100	SP. UNIT LAY ERR.	№ станции/модуля	—	не горит	мигает	стоп	при включении/при сбросе
2101							

<sup>1</sup> В круглых скобках указаны адреса специальных регистров, в которых сохраняется специфическая информация об ошибках.

<sup>2</sup> Состояние (режим) центрального процессора при возникновении ошибки можно установить с помощью параметров. Состояние светодиодного индикатора изменяется соответственно.

<sup>3</sup> Это сообщение об ошибке возможно только в избыточной системе. Ошибка может распознаваться для активной и резервной системы.

<sup>4</sup> Это сообщение об ошибке возможно только в избыточной системе.

<sup>5</sup> Эта ошибка может распознаваться либо в неизбыточной системе, либо в активной части избыточной системы.

## Перечень кодов ошибок (серии QnA и "System Q")

	Причина	Устранение	Отно-сится к:
	Ошибка в одном из установленных модулей. В многопроцессорной системе установлен центральный процессор версии "А".	Замените центральный процессор на процессор версии "В" или выше. Возможно, неисправен специальный модуль, центральный процессор или монтажная шина. Обратитесь в сервисную службу Mitsubishi.	Q, начиная с версии "В"
	Обнаружена ошибка в системной шине.	Возможно, неисправен специальный модуль, центральный процессор или монтажная шина. Обратитесь в сервисную службу Mitsubishi.	Q Rem
	Обнаружена ошибка в главной или расширительной монтажной шине.	Возможно, неисправен специальный модуль, центральный процессор или монтажная шина. Обратитесь в сервисную службу Mitsubishi.	Q, начиная с версии "В"
	При включении обнаружен сбой шины.		
	Модуль управления системой AS92R имеет аппаратную неисправность	Обратитесь в сервисную службу Mitsubishi.	Q4AR
	Кратковременный обрыв электропитания.	Проверьте электропитание.	●/Rem
	Напряжение одного из двух источников питания 5 В в расширительной монтажной шине избыточной системы снизилось ниже 85 % от номинального напряжения.	Проверьте выходное напряжение сетевого блока. Если напряжение не соответствует номинальному, замените сетевой блок.	Q4AR
	Напряжение питания 5 вольт в расширительной монтажной шине снизилось ниже 85 % от номинального напряжения.		
	Напряжение питания 24 В модуля управления AS92R снизилось ниже 85 % от номинального напряжения.		
	1.) Напряжение батареи в центральном процессоре снизилось ниже допустимого значения. 2.) Батарея центрального процессора не соединена с центральным процессором.	1.) Замените батарею. 2.) Если батарея служит для внутренней RAM или функций резервирования, подсоедините провод батареи к модулю центрального процессора.	●
	Напряжение батареи в карте памяти 1 снизилось ниже допустимого предела.	Замените батарею.	●
	Напряжение батареи в карте памяти 2 снизилось ниже допустимого предела.		QnA
	При включении электропитания изменилась информация модуля ввода-вывода. Во время работы модуль ввода-вывода (или специальный модуль) отсоединился от монтажной шины или не был с ней соединен.	Прочтите общую информацию об ошибках на дисплее программатора и проверьте и/или восстановите правильную посадку соответствующих модулей в слоте. Можно также с помощью программатора проверить специальные регистры с SD1400 по SD 1431 и проверить (улучшить) посадку в слоте тех модулей, бит который установлен на "1".	● Rem
	В многопроцессорной системе установлен центральный процессор версии "А".	Замените центральный процессор процессором версии "В".	Q, начиная с версии "В"
	В конфигурации ввода-вывода слоту, в который вставлен модуль прерываний QI60, не присвоен "Интеллектуальный специальный модуль" или "Модуль прерываний".	Приведите конфигурацию ввода-вывода в соответствие с действительным составом системы.	Q, начиная с версии "В"
	Неправильное присвоение адресов в параметрах: Специальному модулю присвоен адрес модуля ввода-вывода (или наоборот). Модулю, не являющемуся модулем центрального процессора, присвоен адрес центрального процессора (или наоборот). Слоту центрального процессора не сопоставлен никакой центральный процессор. Универсальное реле присвоено модулю, который для этого не пригоден.	Установите параметры адресации и согласуйте их с действительной ситуацией.  Сбросьте настройку универсальных реле.	Q Rem
	Неправильное присвоение адресов в параметрах. Специальному модулю присвоен адрес модуля ввода-вывода (или наоборот).	Установите параметры адресации и согласуйте их с действительной ситуацией.	QnA
	В системе имеется более 12 специальных модулей серии "А" (кроме QI60/A1SI61), способных выполнять прерывания для центрального процессора.	Уменьшите количество специальных модулей из серии "А" (кроме Q/60 и A1SI61) до 12 или менее.	Q
	В системе имеется более 12 специальных модулей (кроме AI61), способных выполнять прерывания для центрального процессора.	Уменьшите количество специальных модулей (кроме AI61) до 12 или менее.	QnA

## Перечень кодов ошибок (серии QnA и "System Q")

Перечень кодов ошибок серий QnA и "System Q" (продолжение)

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики
				RUN	ERROR		
2102	SP. UNIT LAY ERR.	№ станции/модуля	—	не горит	мигает	стоп	при включении/при сбросе
2103							
2104							
2105							
2106	SP. UNIT LAY ERR.	№ станции/модуля	—	не горит	мигает	стоп	при включении/при сбросе
2107							
2108							
2109 <sup>2</sup>							

<sup>1</sup> В круглых скобках указаны адреса специальных регистров, в которых сохраняется специфическая информация об ошибках.

<sup>2</sup> Это сообщение об ошибке может возникать только в резервной части избыточной системы.

## Перечень кодов ошибок (серии QnA и "System Q")

Причина	Устранение	Отно- сится к:
В системе имеется более 6 модулей A1SD51S.	Уменьшите количество модулей A1SD51S до 6 или менее.	Q
В системе имеется более 6 модулей последовательной коммуникации (кроме A(1S)J71QC24).	Уменьшите количество модулей последовательной коммуникации (кроме A(1S)J71QC24) до 6 или менее.	QnA Rem
1.) В однопроцессорной системе установлено более одного модуля прерываний Q160 или A1S161. 2.) В многопроцессорной системе одному центральному процессору сопоставлено более одного модуля прерываний Q160/A1S161. 3.) В многопроцессорной системе установлено более одного модуля A1S161.	1.) Установите только один модуль Q160 или A1S161. 2.) Каждому центральному процессору должен быть сопоставлен только один модуль Q160 или A1S161. 3.) Установите только один модуль A1S161. Если в многопроцессорной системе каждому центральному процессору требуется сопоставить модуль прерываний, используйте модуль Q160. Сочетайте один модуль A1S161 максимум с тремя модулями Q160 или применяйте только модули Q160.	Q, начиная с версии "B"
На монтажной шине имеется более одного модуля прерываний Q160/A(1)S161.	Установите только один модуль Q160 или A1S161.	Q
На монтажной шине имеется более одного модуля прерываний A1S61.	Установите только один модуль A1S61.	QnA
Параметры автоматического обновления MELSECNET/MINI, присвоенные некоторым модулям в сети, не соответствуют действительной ситуации.	Сбросьте параметры автоматического обновления MELSECNET/MINI и приведите их в соответствие с действительной ситуацией.	QnA
Превышено максимальное количество специальных модулей, сопоставленных модулю центрального процессора и способных обрабатывать специализированные команды (максимальное количество не должно превышать 1344). (количество установленных AD59 x 5) (количество установленных AD57(S1)/AD58 x 8) (количество установленных AJ71C24(S3/S6/S8 x 10) (количество установленных AJ71UC24 x 10) (количество установленных AJ71C21(S1) x 29) (количество установленных AJ71PT32(S3) x 125) (количество установленных AJ71QC24 (R2, R4) x 29) (количество установленных AJ71ID1 (2)-R4 x 18) (количество установленных AD75 x 12)  Всего > 1344	Уменьшите количество установленных специальных модулей.	QnA
1.) В многопроцессорной системе установлено более четырех модулей MELSECNET/H. 2.) В многопроцессорной системе установлено более четырех модулей Ethernet серии "System Q".	Используйте максимум 4 модуля.	Q, начиная с версии "B"
1.) Установлено более четырех модулей MELSECNET/H. 2.) Установлено более четырех модулей Ethernet серии "System Q". 3.) В сети MELSECNET/H имеются идентичные номера сетей или станций.	1.) Используйте максимум 4 модуля. 2.) Используйте максимум 4 модуля. 3.) Проверьте номера сетей и станций.	Q Rem
1.) В системе установлено более четырех модулей AJ71QLP21 или AJ71QBR11. 2.) В системе установлено более двух модулей AJ71AP21/R21 или AJ71AT21B. 3.) В системе в общей сложности установлено более четырех модулей AJ71QLP21, AJ71QBR11, AJ71AP21/R21 или AJ71AT21. 4.) В сети MELSECNET/10 имеются идентичные номера сетей или станций. 5.) В одно и то же время в MELSECNET (II) или MELSECNET/B имеется более одной мастер-станции или локальной станции.	1.) Используйте максимум 4 модуля. 2.) Используйте максимум 2 модуля. 3.) Используйте в общей сложности максимум 4 модуля. 4.) Проверьте номера сетей и станций. 5.) Проверьте номера станций.	QnA
Головной адрес, присвоенный для сопоставления адресов в параметрах, совпадает с адресом другого модуля.	Сбросьте параметры адресации и согласуйте их с действительной ситуацией.	● Rem
Установлены модули A1SJ71LP21, A1SJ71BR11, A1SJ71AP21, A1SJ71AR21 или A1SJ71AT2, предназначенные для применения в сети процессора A2US. Установлены модули A1SJ71QLP21 или A1SJ71QBR11, предназначенные для применения в сети процессора Q2AS.	Замените эти модули модулями QJ71LP21 или QJ71BR11.	Q
Установлены модули A(1S)J71LP21 или A(1S)J71BR11, предназначенные для применения в сети процессора AnU.	Замените эти модули на модули A(1S)J71QLP21 или A(1S)J71QBR11.	QnA
В избыточной системе активная и резервная система по-разному сконфигурированы в резервном режиме.	Проверьте конфигурацию резервной системы.	Q4AR

## Перечень кодов ошибок (серии QnA и "System Q")

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики
				RUN	ERROR		
2110	SP UNIT ERROR	№ станции/модуля	локализация ошибки программы	не горит/горит	мигает/горит	стоп/продолжение <sup>2</sup>	во время выполнения блока команд FROM/TO
2111							
2112	SP UNIT ERROR	№ станции/модуля	локализация ошибки программы	не горит/горит	мигает/горит	стоп/продолжение <sup>2</sup>	во время выполнения блока команд FROM/TO
2113		FFFFH (всегда)					
2114	SP UNIT ERROR	№ станции/модуля	локализация ошибки программы	мигает/горит	мигает/горит	стоп/продолжение	при выполнении команды
2115							
2116							
2117							
2120	SP. UNIT LAY ERR.	—	—	не горит	мигает	стоп	при включении/при сбросе
2121							
2122							
2124							
2125							
2126							
2150	SP.UNIT VER. ERR.	№ станции/модуля	—	не горит	мигает	стоп	при включении/при сбросе

<sup>1</sup> В круглых скобках указаны адреса специальных регистров, в которых сохраняется специфическая информация об ошибках.

<sup>2</sup> Состояние (режим) центрального процессора при возникновении ошибки можно установить с помощью параметров. Состояние светодиодного индикатора изменяется соответственно.



## Перечень кодов ошибок (серии QnA и "System Q")

	Причина	Устранение	Отно-сится к:
	В команде, обращающейся к общей области памяти в многопроцессорной системе, указан несуществующий центральный процессор.	Прочтите специфическую информацию об ошибках и проверьте (измените) программирование команды.	Q, начиная с версии "B"
	Модуль, к которому обращается команда FROM/TO, не является специальным модулем. Запрашиваемый специальный модуль имеет неполадку.	Прочтите специфическую информацию об ошибках и проверьте (измените) программирование команд FROM и TO. Если специальный модуль неисправен, обратитесь в сервисную службу Mitsubishi.	●
	Модуль, к которому программа обращается с помощью непосредственно адресуемых операндов связи, не является модулем сетевой коммуникации.		
	Запрашиваемый специальный модуль не является специальным модулем или является неправильным специальным модулем.	Прочтите специфическую информацию об ошибках и проверьте (измените) программирование команды.	● Rem
	Не установлены никакие данные имитации специального модуля.	Прочтите специфическую информацию об ошибках и проверьте (обработайте) данные имитации специального модуля.	●
	В команде, в которой должен быть указан другой центральный процессор, указан тот самый процессор, в котором вызывается эта команда.	Прочтите специфическую информацию об ошибках и проверьте (исправьте) программу.	Q, начиная с версии "B"
	В команде, которая обращается к центральному процессору, вызывающему эту команду, указан другой центральный процессор.		
	Применена команда, в которой нельзя указывать специальный модуль, присвоенный другому центральному процессору.		
	В специфической многопроцессорной команде указан недопустимый центральный процессор.		
	Монтажные шины Q[]B и QA1S[]B неправильно расположены.	Проверьте компоновку.	Q Rem
	Центральный процессор вставлен в иной слот, а не в слот центрального процессора (слот от 0 до 2).	Проверьте место расположения центрального процессора.	
	Шина QA1S[] установлена в качестве главной монтажной шины.	В качестве главной монтажной шины используйте Q[]B.	
	Модуль вставлен в 65-й слот или слот с еще более высоким номером. Модуль вставлен в слот, который более не предусмотрен в диапазоне адресов. Модуль занимает адреса входов/выходов, находящиеся вне допустимых 4096 адресов ввода-вывода. Модуль, установленный в качестве 4096-го адреса ввода-вывода, занимает и другие адреса.	Удалите все модули, установленные начиная с 65-го слота. Удалите модуль, вставленный в слот вне выделенного диапазона. Удалите модуль, установленный вне 4096 адресов ввода-вывода. Замените этот модуль модулем, не превышающим 4096 адресов ввода-вывода.	
	Установлен модуль, не распознаваемый центральным процессором Q. Доступ к специальному модулю не возможен.	Используйте модуль, совместимый с центральным процессором Q. Запрашиваемый специальный модуль имеет аппаратную неисправность. Обратитесь в сервисную службу Mitsubishi.	
	1.) В многопроцессорной системе между центральными процессорами имеется пустой слот. 2.) Между двумя процессорами Q установлен другой модуль (например, процессор движения, модуль ввода/вывода).	1.) Между модулями центральных процессоров не должно быть пустых слотов. Пустые слоты допускаются справа от модулей центральных процессоров. 2.) Удалите модуль, установленный между модулями центральных процессоров. Процессор движения должен быть расположен справа от процессоров Q.	
	Специальный модуль, не пригодный для многопроцессорного режима, сопоставлен центральному процессору 2, 3, или 4.	1.) Замените специальный модуль специальным модулем, совместимым с многопроцессорным режимом. 2.) Сопоставьте модуль, не пригодный для многопроцессорного режима, центральному процессору 1.	Q, начиная с версии "B"

## Перечень кодов ошибок (серии QnA и "System Q")

Перечень кодов ошибок серий QnA и "System Q" (продолжение)

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики
				RUN	ERROR		
2200	MISSING PARA.	название дисковогода	—	не горит	мигает	стоп	при включении/при сбросе
2210	BOOT ERROR	название дисковогода	—	не горит	мигает	стоп	при включении/при сбросе
2300	ICM. OPE. ERROR	название дисковогода	—	не горит/ горит	мигает/горит	стоп/ продолжение <sup>2</sup>	при вставлении или вынимании карты памяти
2301							
2302							
2400	FILE SET ERROR	название файла	номер параметра	не горит	мигает	стоп	при включении/при сбросе
2401							
2410	FILE OPE. ERROR	название файла	локализация ошибки программы	не горит/ горит	мигает/горит	стоп/ продолжение <sup>2</sup>	при выполнении команды
2411							
2412							
2413							

<sup>1</sup> В круглых скобках указаны адреса специальных регистров, в которых сохраняется специфическая информация об ошибках.

<sup>2</sup> Состояние (режим) центрального процессора при возникновении ошибки можно установить с помощью параметров. Состояние светодиодного индикатора изменяется соответственно.

## Перечень кодов ошибок (серии QnA и "System Q")

	Причина	Устранение	Отно-сится к:
	В дисковом, указанном с помощью двухпозиционных микропереключателей, нет файла параметров.	Проверьте, правильно ли указаны дисководы в настройках параметров. Сохраните файл параметров в дисковом, заданном в параметрах.	●
	Ошибочное содержимое файла загрузки.	Проверьте настройку параметров.	Q
	В дисковом, указанном с помощью двухпозиционных микропереключателей, нет файла загрузки, хотя переключатель "BOOT" установлен в состояние "включено".	Проверьте, правильно ли указаны дисководы в настройках параметров. Сохраните файл первоначальной загрузки в дисковом, заданном в параметрах.	QnA
	Карта памяти была вынута без предварительной установки выключателя деблокировки карты памяти в положение "включено".	Перед выниманием карты памяти переключите выключатель деблокировки карты памяти в положение "включено".	●
	1.) Карта памяти не сформатирована. 2.) Неправильное состояние формата карты памяти.	1.) Сформируйте карту памяти. 2.) Сформируйте карту памяти заново.	
	Вставлена карта памяти, не пригодная для этого центрального процессора.	Проверьте карту памяти.	
	Сделана попытка автоматически передать данные в стандартную память ROM в центральном процессоре, в котором эта функция не возможна (в файле первоначальной загрузки выбрана автоматическая передача с карты памяти в стандартную ROM, а в качестве действительного источника параметров указана карта памяти).	1.) Активируйте автоматическую запись в стандартную ROM только в тех центральных процессорах, в которых эта функция поддерживается. 2.) Передайте параметры и программы в стандартную ROM с помощью среды программирования. 3.) Выключите автоматическую запись в стандартную ROM и запустите процесс начальной загрузки на основе данных на карте памяти.	Q, начиная с версии "B"
	Не найден файл, указанный в параметрах в файловых настройках контроллера.	Прочтите специфическую информацию об ошибках на дисплее программатора и проверьте, соответствуют ли названия дисковода и файла настройкам параметров. Если необходимо, откорректируйте. Создайте заданный файл.	●
	Параметр Ethernet, действительный для процессора QnA версии "B", использован для процессора QnA, не соответствующего версии "B".	Используйте QnA в версии "B". Сотрите параметр Ethernet.	QnA
	В процессе начальной загрузки или при автоматической записи в стандартную ROM превышена емкость памяти для хранения программы.	– Проверьте и, если необходимо, откорректируйте настройки процесса начальной загрузки. – Сотрите ненужные файлы из памяти для хранения программы. – Выберите в параметрах опцию "Стирать память для хранения программ", чтобы перед загрузкой происходило стирание памяти для хранения программ.	Q, начиная с версии "B"
	В области протокола ошибок не удалось создать файл, указанный в параметрах в настройках RAS контроллера.	Прочтите специфическую информацию об ошибках на дисплее программатора и проверьте соответствие программы настройкам параметров. Если необходимо, откорректируйте настройки. Проверьте оставшуюся свободную память на карте памяти.	●
	Не найден файл, предусмотренный в основной программе.	Прочтите специфическую информацию об ошибках на дисплее программатора и убедитесь в том, что указанная в параметрах программа находится в указанном дисковом. Если необходимо, сделайте коррекции. Создайте указанный файл.	●
	Основная программа не может обращаться к файлу этого типа (файлы комментариев и т. п.).	Прочтите специфическую информацию об ошибках на дисплее программатора и убедитесь в том, что указанная в параметрах программа находится в указанном дисковом. Если необходимо, сделайте коррекции.	
	Основная программа не может обращаться к программному файлу, написанному на языке SFC.		
	В установленный основной программой файл не были записаны данные.	Прочтите специфическую информацию об ошибках на дисплее программатора и убедитесь в том, что указанная в параметрах программа находится в указанном дисковом. Если необходимо, сделайте коррекции. Проверьте, не защищен ли указанный файл от записи.	

## Перечень кодов ошибок (серии QnA и "System Q")

### Перечень кодов ошибок серий QnA и "System Q" (продолжение)

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики		
				RUN	ERROR				
2500	CAN'T EXE. PRG.	название файла	—	не горит	мигает	стоп	при включении/при сбросе		
2501									
2502									
2503									
2504									
3000	PARAMETER ERROR	название файла	номер параметра	не горит	мигает	стоп	при включении/сбросе/переходе STOP → RUN		
3001									
3002		название файла/номер дисковогода					во время выполнения команды END		
3003								название файла	при включении/сбросе/переходе STOP → RUN
3004									
3009	PARAMETER ERROR	название файла/ номер дисковогода	номер параметра	не горит	мигает	стоп	при включении/сбросе/переходе STOP → RUN		
3010									
3012									
3013									

<sup>1</sup> В круглых скобках указаны адреса специальных регистров, в которых сохраняется специфическая информация об ошибках.

## Перечень кодов ошибок (серии QnA и "System Q")

	Причина	Устранение	Отно-сится к:
	Имеется программный файл, который использует операнды вне диапазона, установленного в параметрах операндов.	Прочтите специфическую информацию об ошибках на дисплее программатора и убедитесь в том, что настройки операндов параметров и операндов программного файла соответствуют действительной ситуации. Если необходимо, откорректируйте.	●
	Имеются программные файлы, хотя в параметрах указано "Не имеется".	Измените программные параметры на "Имеются". Сотрите ненужные программы.	
	Ошибочный программный файл. Или содержимое файлов написано не на языке SFC.	Проверьте, имеет ли файл формат ***.QPG и предназначено ли содержимое файла для основной программы.	
	Программный файл не существует.	Проверьте конфигурацию программы.	
	Указано более одной обычной программы или программы управления на языке SFC.	Проверьте параметры и конфигурацию программы.	
	В настройках указателей прерывания указан специальный модуль, который в многопроцессорной системе сопоставлен другому центральному процессору.	1.) Введите начальный адрес области ввода-вывода модуля, не сопоставленному другому центральному процессору. 2.) Сотрите настройки указателей прерывания.	Q, начиная с версии "B"
	Настройки параметров для установки таймеров, контакта RUN-PAUSE, общего адреса указателей, обработки всех данных, количества свободных слотов или настройки системных прерываний находятся вне диапазона, который может использовать центральный процессор.	1.) Прочтите подробную информацию об ошибках на дисплее программатора, проверьте записи в параметрах и, если необходимо, сделайте коррекции. 2.) Если ошибка продолжает иметь место, это указывает на сбой внутренней памяти RAM центрального процессора или карты памяти.	● Rem
	Содержимое параметров уничтожено.		●
	Не существует файл параметров, указанный в параметре "use the following file" для процессора Q.		
	Область для автоматического обмена данными в многопроцессорной системе превышает диапазон имеющихся регистров файлов.	Используйте область регистров файлов, позволяющую обмен данными.	Q, начиная с версии "B"
	Установленное в параметрах операндов количество операндов находится вне диапазона, который может использовать центральный процессор.	1.) Прочтите подробную информацию об ошибках на дисплее программатора, проверьте записи в параметрах и, если необходимо, сделайте коррекции. 2.) Если ошибка продолжает иметь место, это указывает на сбой внутренней памяти RAM центрального процессора или карты памяти.	●
	Файл параметров не может быть использован центральным процессором или не содержит параметры.		
	В многопроцессорной системе один и тот же модуль сопоставлен нескольким центральным процессорам.	Модуль можно сопоставить только одному центральному процессору. Измените конфигурацию ввода-вывода в каждом центральном процессоре многопроцессорной системы.	Q, начиная с версии "B"
	Настроенное количество модулей центральных процессоров отличается от действительного имеющегося количества.	Приведите количество модулей центральных процессоров в соответствие с конфигурацией системы.	
	Параметры многопроцессорной системы в отдельных модулях центральных процессоров отличаются от параметров центрального процессора 1.	Приведите параметры в отдельных модулях центральных процессоров в соответствие с настройками в центральном процессоре 1.	
	Ошибочные настройки автоматического обмена данными в многопроцессорной системе: 1.) В случае битовых операндов выбран начальный адрес, не равный 0 и не кратный 16. 2.) Указан неправильный операнд. 3.) Настроено нечетное количество операндов.	1.) В качестве начального адреса битового операнда укажите либо 0, либо число, кратное 16. 2.) Укажите правильный операнд. 3.) Укажите четное количество операндов.	

## Перечень кодов ошибок (серии QnA и "System Q")

Перечень кодов ошибок серий QnA и "System Q" (продолжение)

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики
				RUN	ERROR		
3100							
3101							
3102	LINK PARA. ERROR	название файла	номер параметра	не горит	не горит	стоп	при включении/сбросе/переходе STOP → RUN
3103							
3104							
3105							

<sup>1</sup> В круглых скобках указаны адреса специальных регистров, в которых сохраняется специфическая информация об ошибках.

## Перечень кодов ошибок (серии QnA и "System Q")

Причина	Устранение	Отно-сится к:
Модуль MELSECNET/H с начальным адресом области ввода-вывода, указанным в параметрах, присвоен другому центральному процессору в многопроцессорной системе.	Сотрите параметры для модуля MELSECNET/H, сопоставленного другому центральному процессору, и укажите начальный адрес правильного модуля.	Q, начиная с версии "B"
Параметры сетевой коммуникации обычной станции переданы в управляющую станцию, или наоборот.	Выполните сброс центрального процессора.	
Количество установленных модулей отличается от количества, заданного в параметрах для MELSECNET/10(H). Начальный адрес установленных модулей отличается от начального адреса, заданного в параметрах для MELSECNET/10(H). Не все данные в параметрах могут быть считаны. При включенном напряжении изменен тип станции, подключенной к сети MELSECNET/10 (H). (Чтобы система распознала измененный тип, ее необходимо перевести из состояния RESET в состояние RUN).	1.) После коррекции запишите сетевые параметры заново. 2.) Если ошибка продолжает иметь место и после исправления, обратитесь в сервисную службу Mitsubishi.	Q
Не были записаны параметры сети, хотя центральный процессор QnA является управляющей станцией или мастер-станцией.	1.) После коррекции запишите сетевые параметры заново. 2.) Если ошибка продолжает иметь место и после исправления, обратитесь в сервисную службу Mitsubishi.	QnA
1.) В модуле MELSECNET/H с номером станции 0 сделаны настройки для коммуникации между контроллерами. 2.) В модуле MELSECNET/H, номер станции которого не равен 0, сделаны настройки для удаленной мастер-станции.	Измените тип станции или номер станции.	Q, начиная с версии "B"
Номер коммуникационной сети, заданный в параметре, не совпадает с номером установленной сети. Начальный адрес установленных модулей ввода-вывода отличается от начального адреса, заданного в параметрах. Класс коммуникационной сети, заданный в параметре, не совпадает с классом действительно установленной сети. Имеется ошибка в параметрах обновления MELSECNET/10(H).	Приведите параметры в соответствии с установленной коммуникационной сетью.	●
При проверке параметров сети модуля сетевой коммуникации возникла ошибка.	1.) После коррекции запишите сетевые параметры заново. 2.) Если ошибка продолжает иметь место и после исправления, обратитесь в сервисную службу Mitsubishi.	●
Модуль Ethernet с начальным адресом области ввода-вывода, указанным в параметрах, присвоен другому центральному процессору в многопроцессорной системе.	Сотрите параметры для модуля Ethernet, сопоставленного другому центральному процессору, и укажите начальный адрес правильного модуля.	Q, начиная с версии "B"
Хотя в параметрах задан как минимум один модуль Ethernet, ни один модуль не установлен. Начальный адрес установленных модулей ввода-вывода отличается от начального адреса, заданного в параметрах Ethernet.		● Rem
AJ71QE71 не занимает область ввода-вывода, заданную в параметрах. Присвоенные адреса ввода-вывода перекрываются. Настройки параметров отличаются от параметров, загруженных из AJ71QE71. Сумма параметров Ethernet и соответствующих команд установлена более чем на 5.	1.) После коррекции запишите сетевые параметры заново. 2.) Если ошибка продолжает иметь место и после исправления, обратитесь в сервисную службу Mitsubishi.	QnA
Сетям Ethernet и MELSECNET/10 присвоены одни и те же номера сетей. Данные в параметрах номера сети, номера станции или номера группы превышают допустимый диапазон. Заданный адрес ввода-вывода превышает диапазон, допустимый для центрального процессора.		● Rem
Модуль CC-Link с начальным адресом области ввода-вывода, указанным в параметрах, присвоен другому центральному процессору в многопроцессорной системе.	Сотрите параметры модуля CC-Link, сопоставленного другому центральному процессору, и укажите правильный модуль.	Q, начиная с версии "B"
Хотя в параметрах указан как минимум один модуль CC-Link, ни один модуль не установлен. Начальный адрес области ввода-вывода, указанный в общих параметрах, отличается от установленного модуля ввода-вывода. Класс станции CC-Link, заданный в параметре, не совпадает с классом действительно установленной станции.	1.) После коррекции запишите сетевые параметры заново. 2.) Если ошибка продолжает иметь место и после исправления, обратитесь в сервисную службу Mitsubishi.	● Rem
Ошибочное содержимое параметров Ethernet.	После коррекции запишите сетевые параметры заново.	QnA

## Перечень кодов ошибок (серии QnA и "System Q")

Перечень кодов ошибок серий QnA и "System Q" (продолжение)

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики
				RUN	ERROR		
3106	LINK PARA. ERROR	название файла/ дисковода	номер параметра	не горит	мигает	стоп	во время выполнения команды END
3107		название файла					при включении/сбросе/переходе STOP → RUN
		название файла					
3200	SFC PARA. ERROR	название файла	номер параметра	не горит	мигает	стоп	переход STOP → RUN
3201							
3202							
3203							
3300	SP. PARA. ERROR	название файла	номер параметра	не горит	мигает	стоп	при включении/сбросе/переходе STOP → RUN
3301		название файла	номер параметра				во время выполнения команды END
3302							при включении/сбросе/переходе STOP → RUN
3303		название файла/ дисковода	номер параметра				при включении/сбросе/переходе STOP → RUN
3400	REMOTE PASS. ERROR	—	—	не горит	мигает	стоп	при включении/сбросе/переходе STOP → RUN
3401							
4000	INSTRCT CODE. ERR.	локализация ошибки программы	—	не горит	мигает	стоп	при включении/сбросе/переходе STOP → RUN
4001							
4002							
4003							
4004							

<sup>1</sup> В круглых скобках указаны адреса специальных регистров, в которых сохраняется специфическая информация об ошибках.



## Перечень кодов ошибок (серии QnA и "System Q")

	Причина	Устранение	Отно- сится к:
	Область для обновления CC-Link превышает диапазон имеющихся регистров файлов	Используйте для обновления область регистров файлов, позволяющую обновление в соответствии с настройками.	Q, начиная с версии "B"
	Имеется ошибка в параметрах обновления CC-Link.	Проверьте параметрирование.	Q/Rem
	Ошибочное содержимое параметров CC-Link.	Проверьте параметрирование.	●
	Неправильное содержимое параметров.	После коррекции запишите параметры заново.	●
	Неправильная информация атрибутов блоков программы на языке SFC.		
	Установленное в параметрах количество шаговых маркеров меньше количества шаговых маркеров, используемых программой.		
	Настройка режима обработки (Execution type) в параметрах для программы на языке SFC иная, чем "Scan execution type".		
	Головной адрес для специального модуля, присвоенный средой GX Configurator, не совпадает с действительным адресом ввода-вывода.	Проверьте параметрирование	Q/Rem
	Настройка обновления для специального модуля превышает диапазон имеющихся регистров файлов.	Используйте для обновления область регистров файлов, позволяющую обновление в соответствии с настройками.	Q, начиная с версии "B"
	Настройки обновления специального модуля находятся вне допустимого диапазона.	Проверьте параметрирование	
	Неправильные настройки обновления специального модуля.		
	В многопроцессорной системе сделаны настройки для специального модуля, сопоставленного другому центральному процессору.	Сотрите настройки для модуля, сопоставленного другому центральному процессору, и запараметрируйте модуль в том центральном процессоре.	●
	В файле паролей удаленного доступа головной адрес целевого модуля не равен 0 <sub>h</sub> или 0FF0 <sub>h</sub> .	Измените головной адрес целевого модуля на 0 <sub>h</sub> или 0FF0 <sub>h</sub> .	Q, начиная с версии "B"
	Головным адресом в файле паролей удаленного доступа задан неправильный слот. Возможные причины: – Модуль не установлен. – Установлен иной модуль, кроме QJ71C24(-R2) или модуля Ethernet серии "System Q". – Установлен QJ71C24(-R2) или модуль Ethernet серии "System Q" версии "A".	Вставьте в слот, заданный головным адресом в файле паролей удаленного доступа, модуль QJ71C24(-R2) версии "B" или модуль Ethernet серии "System Q", версия "B".	
	Происходит обращение к модулю QJ71C24(-R2), версия "B", или модулю Ethernet серии "System Q", версия "B", сопоставленному другому центральному процессору.	Обращайтесь к правильному модулю и сотрите настройки пароля удаленного доступа.	
	В программе имеется код команды, который не удалось декодировать.	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.	●
	Программа, не являющаяся основной программой, содержит специализированную команду, предназначенную для основной программы.		
	Команда имеет неправильное название.		● Rem
	Команда обращается к неправильному адресу операнда.		●
	Команда обращается к операнду, который нельзя использовать.		

## Перечень кодов ошибок (серии QnA и "System Q")

Перечень кодов ошибок серий QnA и "System Q" (продолжение)

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики
				RUN	ERROR		
4010	MISSING END INS.	локализация ошибки программы	—	не горит	мигает	стоп	при включении/сбросе/переходе STOP → RUN
4020	CAN'T SET (P)	локализация ошибки программы	—	не горит	мигает	стоп	при включении/сбросе/переходе STOP → RUN
4021							
4030	CAN'T SET (I)	локализация ошибки программы	—	не горит	мигает	стоп	при включении/сбросе/переходе STOP → RUN
4100	OPERATION ERROR	локализация ошибки программы	—	не горит/ горит	мигает/горит	стоп/ продолжение <sup>2</sup>	при выполнении команды
4101			—				
4102		программа	локализация ошибки программы				
		локализация ошибки программы	—				
4103		локализация ошибки программы	—				
4107		программа	локализация ошибки программы				
		локализация ошибки программы	—				
4108	локализация ошибки программы	—					
4200	FOR NEXT ERROR	локализация ошибки программы	—	не горит	мигает	стоп	при выполнении команды
4201							
4202							
4203							
4210	CAN'T EXECUTE (P)	локализация ошибки программы	—	не горит	мигает	стоп	при выполнении команды
4211							
4212							
4213							

<sup>1</sup> В круглых скобках указаны адреса специальных регистров, в которых сохраняется специфическая информация об ошибках.

<sup>2</sup> Состояние (режим) центрального процессора при возникновении ошибки можно установить с помощью параметров. Состояние светодиодного индикатора изменяется соответственно.

## Перечень кодов ошибок (серии QnA и "System Q")

	Причина	Устранение	Отно-сится к:
	В программе нет команды END (FEND)	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (исправьте) указанные файлы.	●
	Общее количество внутренних указателей файла, используемых программой, превышает количество, установленное в параметрах.	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.	●
	Перекрываются адреса общих указателей, используемые соответствующими файлами.		
	Перекрываются адреса присвоенных указателей, используемые соответствующими файлами.		
	Содержащиеся данные не могут обрабатываться соответствующей командой.	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.	●
	Указанные адреса данных, которые должна обрабатывать программа, или сохраненные данные или константы операндов, используемых командами, находятся вне диапазона адресов, который можно использовать.		
	В многопроцессорной системе специализированная команда сетевой коммуникации обращается к модулю, сопоставленному другому центральному процессору.	Сотрите в программе команду, вызвавшую ошибку, и обращайтесь к этому модулю из того центрального процессора, которому сопоставлен этот модуль.	Q, начиная с версии "B"
	Неправильный номер сети или станции, к которому обращается специализированная команда сетевой коммуникации.	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.	● Rem
	Неправильная конфигурация специализированной команды PID.		●
	В одном центральном процессоре многопроцессорной системы выполнено более 32 специфических многопроцессорных команд.	Используйте блокирующий битовый операнд, показывающий выполнение команды - чтобы предотвратить одновременное выполнение более 32 команд для многопроцессорной системы.	Q, начиная с версии "B"
	Команда CC-Link выполняется более 64 раз.	Выполняйте команду CC-Link не более 64 раз.	QnA
	При выполнении команды CC-Link не были введены параметры CC-Link.	Перед выполнением команды CC-Link установите параметры.	
	После команды FOR не выполняется команда NEXT, или имеется меньшее количество команд NEXT, чем команд FOR.	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (исправьте) указанный шаг программы.	●
	Выполняется команда NEXT, хотя не была выполнена команда FOR, или имеется больше команд NEXT, чем команд FOR.		
	Запрограммировано более 16 уровней вложения.	Уменьшите количество уровней вложения, так чтобы их было менее 17.	●
	Выполняется команда BREAK, хотя не была выполнена команда FOR.	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.	●
	Выполняется команда CALL, однако на используемом указателе нет подпрограммы.	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.	
	В выполненной подпрограмме нет команды RET.		
	Команда RET находится перед командой FEND в основной программе.		
	Запрограммировано более 16 уровней вложения.	Уменьшите количество уровней вложения, так чтобы их было менее 17.	

## Перечень кодов ошибок (серии QnA и "System Q")

Перечень кодов ошибок серий QnA и "System Q" (продолжение)

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики
				RUN	ERROR		
4220	CAN'T EXECUTE (I)	локализация ошибки программы	—	не горит	мигает	стоп	при выполнении команды
4221							
4223							
4230	INST. FORMAT ERR	локализация ошибки программы	—	не горит	мигает	стоп	при выполнении команды
4231							
4235							
4300	EXTEND INST. ERR.	локализация ошибки программы	—	не горит/ горит	мигает/горит	стоп/ продолжение <sup>2</sup>	при выполнении команды
4301							
4400	SFCP. CODE ERROR	локализация ошибки программы	—	не горит	мигает	стоп	STOP → RUN
4410	CAN'T SET (BL)	локализация ошибки программы	—	не горит	мигает	стоп	STOP → RUN
4411							
4420	CAN'T SET (S)	локализация ошибки программы	—	не горит	мигает	стоп	STOP → RUN
4421							
4422							
4500	SFCP. FORMAT ERR.	локализация ошибки программы	—	не горит	мигает	стоп	STOP → RUN
4501							
4502							
4503							
4504							
4600	SFCP. OPE. ERROR	локализация ошибки программы	—	не горит/ горит	мигает/горит	стоп/ продолжение <sup>2</sup>	при выполнении команды
4601							
4602							
4610	SFCP. EXE. ERROR	локализация ошибки программы	—	горит	горит	продолжение	STOP → RUN
4611							

<sup>1</sup> В круглых скобках указаны адреса специальных регистров, в которых сохраняется специфическая информация об ошибках.

<sup>2</sup> Состояние (режим) центрального процессора при возникновении ошибки можно установить с помощью параметров. Состояние светодиодного индикатора изменяется соответственно.

## Перечень кодов ошибок (серии QnA и "System Q")

	Причина	Устранение	Отно-сится к:
	<p>Сделан ввод прерывания, однако соответствующий указатель прерывания не обнаружен.</p> <p>В выполненной программе прерывания нет команды RET.</p> <p>Команда IRET в основной программе находится перед командой FEND.</p>	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.	●
	<p>После команды CHKCR не выполняется команда CHKEND. Имеется одинаковое количество команд CHK и CHKEND.</p> <p>Команды IX и IXEND не запрограммированы совместно. Имеется одинаковое количество команд IX и IXEND.</p> <p>Контрольные условия команды CHK недействительны, или команда CHK применяется в программе с пониженной скоростью обработки.</p>	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.	●
	<p>Неправильное написание управляющей команды для мастер-модуля MELSECNET/mini-S3.</p> <p>Неправильное написание управляющей команды AD57/AD58.</p>	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.	QnA
	В программе на языке SFC нет команды SFCR или SFCPEND.	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.	●
	Используемые программой на языке SFC адреса блоков находятся вне диапазона адресов.		●
	Перекрываются адреса блоков в программе на языке SFC.		●
	Номер шага в программе на языке SFC превышает 255.	Уменьшите количество шагов.	●
	Количество всех шагов во всех программах на языке SFC превышает допустимое значение.		●
	Перекрывается нумерация шагов в программе на языке SFC.		●
	Количество команд BLOCK и BEND в программе на языке SFC не отвечает соотношению 1:1.	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.	●
	Неправильная взаимная конфигурация команд STEP*, TRAN*, TSET и SEND в программе на языке SFC.		●
	В программном блоке на языке SFC нет команды STEP*.		●
	Не существует шаг, к которому обращается команда TSET в программе на языке SFC.	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.	●
	Не существует шаг, к которому обращается команда TAND в программе на языке SFC.		●
	Программа на языке SFC содержит данные, обработка которых не возможна.	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы. Программа запускается на шаге инициализации.	●
	Превышается установленная в программе SFC область операндов.		●
	В очередности выполнения шагов в программе на языке SFC команда END предшествует команде START.		●
	Неправильная информация активного шага для возобновления обработки программы на языке SFC.	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.	●
	Во время возобновления обработки программ на языке SFC выключатель с ключом был переведен из положения "RUN" в положение "RESET".		●

## Перечень кодов ошибок (серии QnA и "System Q")

Перечень кодов ошибок серий QnA и "System Q" (продолжение)

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики
				RUN	ERROR		
4620	BLOCK EXE. ERROR	локализация ошибки программы	—	не горит	мигает	стоп	при выполнении команды
4621							
4630	STEP EXE. ERROR	локализация ошибки программы	—	не горит	мигает	стоп	при выполнении команды
4631							
4632							
4633							
5000	WDT ERROR	время (настройка)	время (фактическое измеренное значение)	не горит	мигает	стоп	непрерывно
5001							
5010	PRG. TIME OVER	время (настройка)	время (фактическое измеренное значение)	горит	горит	продолжение	непрерывно
5011							
6000	PRG. VERIFY ERR. <sup>2</sup>	название файла	—	не горит	мигает	стоп	непрерывно
6010	MODF VERIFY ERR. <sup>2</sup>	—	—	горит	горит	продолжение	непрерывно
6100	TRK. MEMORY ERR. <sup>3</sup>	—	—	горит	горит	продолжение	при включении/сбросе/переходе STOP → RUN
6101							при выполнении команды END
6200	CONTROL EXE. <sup>4</sup>	основание переключения	—	горит	не горит	продолжение	непрерывно
6210	CONTROL WAIT. <sup>2</sup>	основание переключения	—	горит	не горит	продолжение	непрерывно
6220	CAN'T EXE CHANGE <sup>4</sup>	основание переключения	—	горит	горит	продолжение	непрерывно
6221							
6222							

<sup>1</sup> В круглых скобках указаны адреса специальных регистров, в которых сохраняется специфическая информация об ошибках.

<sup>2</sup> Это сообщение об ошибке может возникать только в резервной части избыточной системы.

<sup>3</sup> Это сообщение об ошибке возможно только в избыточной системе. Ошибка может распознаваться для активной и резервной системы.

<sup>4</sup> Это сообщение об ошибке может возникать только в активной части избыточной системы.

## Перечень кодов ошибок (серии QnA и "System Q")

	Причина	Устранение	Отно-сится к:
	Сделана попытка заново запустить уже запущенный блок программы на языке SFC.		●
	Сделана попытка запуска несуществующего блока программы на языке SFC.		
	Сделана попытка заново запустить уже запущенный блок программы на языке SFC.	Прочтите общую информацию об ошибках на дисплее программатора и проверьте (откорректируйте) указанный шаг программы.	●
	Сделана попытка запуска несуществующего блока программы на языке SFC.		
	В блоках программы на языке SFC одновременно выполняется слишком много шагов.		
	В блоках всех программ на языке SFC одновременно выполняется слишком много шагов.		
	Время цикла программы с режимом обработки "Initial execution type" превышает время сторожевого таймера для контроля программ этого типа, настроенное в параметрах RAS контроллера.	Прочтите специфическую информацию об ошибках на дисплее программатора и проверьте (уменьшите) настроенное время цикла.	●
	Время цикла программы превышает контрольное время сторожевых таймеров, настроенное в параметрах RAS контроллера.		
	Время цикла программы в замедленном режиме обработки ("Low speed scan") превышает постоянное время цикла, настроенное в параметре RAS контроллера.	Проверьте и измените постоянное время цикла или время выполнения программы в замедленном режиме ("low speed scan").	●
	Время цикла программы с режимом обработки "Low speed scan type" превышает время сторожевого таймера для контроля программ этого типа, настроенное в параметрах RAS контроллера.	Прочтите специфическую информацию об ошибках на дисплее программатора и проверьте (уменьшите) настроенное время цикла.	
	Активная и резервная часть избыточной системы имеют различающиеся программы и параметры.	Согласуйте программы и параметры обеих систем.	●
	Активная и резервная часть избыточной системы имеют различающиеся рабочие режимы.	Эксплуатируйте обе системы в одном и том же режиме.	
	Во время запуска контроллера обнаружен сбой в согласующей памяти центрального процессора.	Аппаратная неисправность центрального процессора. Обратитесь в сервисную службу Mitsubishi. Замените сначала центральный процессор резервной системы, а затем центральный процессор активной системы.	●
	Во время квитирования связи при слежении центральный процессор обнаружил ошибку.	Проверьте состояние других станций.	
	Резервная часть избыточной системы переключена в состояние активной системы.	Проверьте состояние активной системы.	Q4AR
	Активная часть избыточной системы переключена в состояние резервной системы.		
	Резервную систему не удалось переключить из активного в резервное состояние.	Проверьте состояние резервной системы.	●
	Переключение заблокировано из-за сбоя модуля переключения шины.	Аппаратная неисправность модуля переключения. Обратитесь в сервисную службу Mitsubishi.	
	Переключение заблокировано, так как мастер-станция удаленного периферийного устройства, работающая в мультиплексном режиме, установлена в резервной системе.	Проверьте настройки удаленного периферийного устройства.	

## Перечень кодов ошибок (серии QnA и "System Q")

Перечень кодов ошибок серий QnA и "System Q" (продолжение)

Код ошибки (SD0) <sup>1</sup>	Сообщение об ошибке	Общая информация (SD5...15) <sup>1</sup>	Специфич. информация (SD16...26) <sup>1</sup>	Состояние светодиода		Сост. центр. проц.	Момент диагностики
				RUN	ERROR		
7000	MULT CPU DOWN	номер станции номер модуля	—	не горит	мигает	стоп	непрерывно
7002							при включении/при сбросе
7003							при включении/при сбросе
7010	MULTI EXE. ERROR	номер станции номер модуля	—	не горит	мигает	стоп	при включении/при сбросе
7020	MULTI CPU ERROR		—	горит	горит	продолжение	непрерывно
9000	F**** <sup>2</sup>	локализация ошибки программы	№ маркера ошибки	горит	не горит	продолжение	при выполнении команды
				горит светодиод "USER"			
9010	<CHK> ERR ***_*** <sup>3</sup>	локализация ошибки программы	№ ошибки	горит	не горит	продолжение	при выполнении команды
				горит светодиод "USER"			
9020	BOOT OK	—	—	не горит	мигает	стоп	при включении/сбросе
10000	CONT.UNIT ERROR	—	—	—	—	—	—

<sup>1</sup> В круглых скобках указаны адреса специальных регистров, в которых сохраняется специфическая информация об ошибках.

<sup>2</sup>\*\*\*\* указывают на распознанный номер маркера ошибки.

<sup>3</sup>\*\*\* указывают на контакт и номер проверочного контура.



## Перечень кодов ошибок (серии QnA и "System Q")

	Причина	Устранение	Отно-сится к:
	<p>1.) Возникла ошибка в модуле центрального процессора, в котором выбрана настройка, предусматривающая останов всех центральных процессоров многопроцессорной системы при ошибке этого центрального процессора.</p> <p>2.) В многопроцессорной системе установлен центральный процессор Q версии "А".</p>	<p>1.) Продолжите поиск ошибки на соответствующем модуле центрального процессора.</p> <p>2.) В многопроцессорной системе можно применять только процессоры "Q", начиная с версии "В".</p>	Q, начиная с версии "В"
	<p>При включении напряжения центральный процессор 1 в многопроцессорной системе остановлен из-за ошибки. В результате этого другие центральные процессоры не могут запуститься. Это сообщение об ошибке возникает в центральном процессоре 2, 3 и 4.</p>	<p>Продолжите поиск ошибок в центральном процессоре 1.</p>	
	<p>1.) При установлении связи в многопроцессорной системе отсутствует реакция целевого процессора.</p> <p>2.) В многопроцессорной системе установлен центральный процессор Q версии "А".</p>	<p>1.) Выполните сброс центрального процессора. Если после этого ошибка возникает снова, то, вероятно, имеется аппаратная неисправность в центральном процессоре. В этом случае обратитесь в сервисную службу Mitsubishi.</p> <p>2.) В многопроцессорной системе можно применять только процессоры "Q", начиная с версии "В".</p>	
	<p>В многопроцессорной системе при установлении связи не последовало реакции от целевого процессора.</p>	<p>Выполните сброс центрального процессора. Если после этого ошибка возникает снова, то, вероятно, имеется аппаратная неисправность в центральном процессоре. В этом случае обратитесь в сервисную службу Mitsubishi.</p>	
	<p>1.) В многопроцессорной системе неисправен один из центральных процессоров.</p> <p>2.) В многопроцессорной системе установлен центральный процессор Q версии "А" (это сообщение об ошибке возникает в других центральных процессорах (Q, начиная с версии "В")).</p> <p>3.) Центральный процессор 2, 3 или 4 сброшен (это сообщение возникает в центральном процессоре, который был сброшен).</p>	<p>1.) Проанализируйте информацию об ошибках. Замените неисправный центральный процессор.</p> <p>2.) В многопроцессорной системе можно применять только процессоры "Q", начиная с версии "В". Замените центральный процессор (версии "А") процессором версии "В" или выше.</p> <p>3.) Сбросьте центральный процессор 1 для сброса всей многопроцессорной системы.</p>	Q, начиная с версии "В"
	<p>Возникла ошибка в модуле центрального процессора, в котором выбрана настройка, предусматривающая останов всех центральных процессоров многопроцессорной системы при наличии ошибки в этом процессоре. (Это сообщение об ошибке появляется в центральных процессорах, остановленных этим процессором).</p>	<p>Продолжите поиск ошибки на соответствующем модуле центрального процессора.</p>	
	<p>Маркер ошибки F установлен в состояние "включен".</p>	<p>Прочтите специфическую информацию об ошибках на дисплее программатора и проверьте (исправьте) программу на основе номера установленного маркера ошибки.</p>	●
	<p>Команда СНК обнаружила ошибку.</p>		
	<p>Сохранение данных в стандартной памяти ROM завершено без ошибок. Мигает также светодиод "BOOT".</p>	<p>В качестве адреса хранения для данных первоначальной загрузки выберите в параметрах стандартную ROM, а затем выключите и снова включите электропитание, чтобы загрузить данные первоначальной загрузки из стандартной ROM.</p>	Q, начиная с версии "В"
	<p>В многопроцессорной системе возникла ошибка в ином процессоре кроме процессора Q (например, процессоре движения).</p>	<p>Продолжите поиск ошибки на соответствующем модуле центрального процессора.</p>	

### 13.3 Перечень кодов ошибок серии "А" (кроме AnA и AnAS)

В следующей таблице дан обзор возможных кодов ошибок вместе с соответствующими текстовыми сообщениями, а также указаны их возможные причины и способы устранения. Коды ошибок записываются в специальный регистр D9008, а номера шагов, в которых возникла ошибка, в специальные регистры D9010 и D9011. В этой таблице перечислены только сообщения об ошибках, относящиеся к центральным процессорам AnN, AnU, AnS, A3M и A2C.

Сообщение об ошибке	Содержимое рег. D9008	Сост. центр. процессора	Причина	Устранение
INSTRCT. CODE ERR" Ошибочная команда (проверка происходит во время выполнения программы)	10	стоп	В программе содержится команда, которую не может обрабатывать центральный процессор. Вставлен чип EPROM с ошибочной программой. Содержимое памяти было изменено. Запрограммирована команда PR или IRET.	Считать ошибочный шаг программы с помощью программатора и исправить строку программы. При ошибочной программе в EPROM соответственно изменить программу в EPROM или заменить чип памяти.
PARAMETER ERROR" Ошибочный параметр (проверка происходит при включении, сбросе, переключении между состояниями STOP и RUN или PAUSE и RUN)	11	стоп	Установлено слишком большое значение для емкости памяти центрального процессора. Параметры центрального процессора неправильно установлены или изменились под действием помех (память параметров стерта). Не вставлена память RAM (в случае процессора A1 или ATN).	Проверить, правильно ли чип памяти закреплен в цоколе. Проверить область параметров с помощью программатора и исправить ошибки. Если необходимо, заново записать параметры в центральный процессор.
MISSING END INS." Отсутствует команда END (проверка происходит после установки маркера M9056 или M9057 или переключения между состояниями STOP и RUN или PAUSE и RUN)	12	стоп	В основной программе нет команды END. В подпрограмме нет команды END (если подпрограмма определена с помощью параметра).	Вставить команду END в конце программы/ подпрограммы.
CAN'T EXECUTE (P)" Переход программы не выполним (проверка происходит при выполнении одной из следующих команд: CJ, SCJ, JMP, CALLP, FOR/NEXT, а также после переключения между STOP и RUN или PAUSE и RUN)	13	стоп	Цель перехода, к которой обращаются команды CJ, SCJ, CALL, CALLP или JMP, имеет ошибку или отсутствует. В программе имеется команда CHG, однако нет подпрограммы. Запрограммирована (и выполнена) команда RET, хотя в программе нет команды CALL. Цель перехода, к которой обращаются команды CJ, SCJ, CALL, CALLP или JMP, находится после команды END. Количество команд FOR не соответствует количеству команд NEXT. Цель перехода команды JMP, запрограммированной между командами FOR и NEXT, находится вне цикла FOR-NEXT. Цель перехода, заданная в команде JMP, не находится в пределах подпрограммы перед командой RET. Цель перехода команды JMP расположена в шаге программы или в подпрограмме, находящимся (находящейся) в цикле FOR-NEXT. Команда STOP выполнена в программе прерывания, подпрограмме или внутри цикла FOR-NEXT.	Считать ошибочный шаг программы с помощью программатора и исправить строку программы (например, вставить команду перехода или изменить цель перехода).

## Перечень кодов ошибок серии "А" (кроме AnA и AnAS)

### Перечень кодов ошибок серии "А" (кроме AnA и AnAS), продолжение

Сообщение об ошибке	Содержимое рег. D9008	Сост. центр. процессора	Причина	Устранение
<p style="text-align: center;">CHK FORMAT ERR"</p> <p style="text-align: center;">Ошибка в формате CHK (проверка происходит после переключения STOP/RUN или PAUSE/RUN)</p>	14	стоп	<p>В блоке команд CHK (включая NOP) находятся иные команды кроме LDX, LDIX, ANDX и ANIX. Программа содержит более одной команды CHK. Блок CHK содержит более 150 входных контактов (контрольных условий). Адрес входной команды X в блоке CHK превышает максимальное значение. Над блоком команд CHK нет команды CJ с входным условием. Адрес операнда D1 в команде CHK D1 D2 не соответствует адресу операнда над командой CJ. Указатель P254 присвоен не началу блока команд CHK.</p>	<p>Проверить блок команд CHK в программе на возможные причины ошибок в соответствии изложенными рядом пунктами и сделать необходимые исправления.</p>
<p style="text-align: center;">CAN'T EXECUTE (1)"</p> <p style="text-align: center;">выполнение не возможно (проверка происходит при выполнении прерывания)</p>	15	стоп	<p>Адрес указателя прерывания (I), на основе которого требуется обратиться к модулю прерываний, ошибочен или встречается в программе многократно. В программе прерывания нет команды IRET. Команда IRET находится не в программе прерывания, а в иной части программы.</p>	<p>Проверить, имеется ли программа прерывания, которая обращается к данному модулю прерываний. Стереть указатель прерывания, запрограммированный два раза (если таковой имеется). Проверить, не отсутствует ли в программе прерывания команда IRET. Если необходимо, вставить эту команду. Проверить, не встречается ли команда IRET вне программы прерывания, и стереть соответствующее место в программе.</p>
<p style="text-align: center;">CASSETTE ERROR"</p> <p style="text-align: center;">Доступ к кассете памяти не возможен</p>	16	стоп	<p>В модуле центрального процессора нет кассеты памяти или кассета неисправна.</p>	<p>Вставить или заменить кассету памяти (при отключенном напряжении питания).</p>
<p style="text-align: center;">ROM-ERROR"</p> <p style="text-align: center;">Ошибка EPROM (проверка происходит после включения и сброса)</p>	17	стоп	<p>Параметры и основная программа сохранены в применяемой EPROM с ошибками. Неисправно запоминающее устройство EPROM.</p>	<p>Заново передать программу и параметры в EPROM или заменить память EPROM, если она неисправна.</p>
<p style="text-align: center;">MEMORY PROTECT ERROR"</p> <p style="text-align: center;">Ошибка из-за защиты от записи (проверка происходит после включения и сброса)</p>	18	стоп	<p>Во время обращения центрального процессора к программе, сохраненной в EPROM, включена защита от записи (микровыключатель находится в положении "ON").</p>	<p>Переключить выключатель защиты от записи (микровыключатель в центральных процессорах некоторых типов) в положение "OFF".</p>
<p style="text-align: center;">RAM ERROR"</p> <p style="text-align: center;">Ошибка при обращении к памяти RAM (проверка происходит после включения, сброса и установки M9084 в режиме "стоп")</p>	20	стоп	<p>Центральный процессор самостоятельно проверяет правильность выполнения процессов записи в память данных и считывания из памяти данных. При обращении возникла ошибка.</p>	<p>Обратиться в сервисную службу Mitsubishi.</p>
<p style="text-align: center;">OPE. CIRCUIT ERR"</p> <p style="text-align: center;">Неисправность в рабочем контуре (проверка происходит после включения и сброса)</p>	21	стоп	<p>Неправильно работает контур, отвечающий за обработку программы в центральном процессоре.</p>	<p>Обратиться в сервисную службу Mitsubishi.</p>

## Перечень кодов ошибок серии "А" (кроме AnA и AnAS)

### Перечень кодов ошибок серии "А" (кроме AnA и AnAS), продолжение

Сообщение об ошибке	Содержимое рег. D9008	Сост. центр. процессора	Причина	Устранение
WDT ERROR (1) Ошибка WDT 1 (проверка происходит после выполнения команды END)	22	стоп	Время цикла программы превышает время контроля на ошибки с помощью сторожевого таймера. Время цикла программы сильно увеличилось. Время цикла увеличилось из-за кратковременного падения напряжения, произошедшего во время прогона программы.	Проверить и заново рассчитать время цикла программы. Время цикла программы можно уменьшить с помощью команд CJ и т. п.. Если в структуре программы ошибок нет, следует с помощью программатора просмотреть содержимое специального регистра D9005. Если его значение не равно 0, то это означает, что напряжение питания недостаточное. В этом случае следует проверить электропитание и устранить причину потери напряжения.
SUB-CPU ERROR Ошибка подчиненного центрального процессора	23	стоп	Подчиненный центральный процессор неисправен или заблокирован.	Обратиться в сервисную службу Mitsubishi.
END NOT EXECUTE Не удалось выполнить команду END (проверка происходит во время выполнения команды END)	24	стоп	Во время выполнения команды END процессор из-за воздействия помех (наведенных напряжений и т. п.) считал другой код команды. Под действием помех команда END преобразовалась в другую команду.	Выполнить сброс и заново перевести центральный процессор в режим RUN. Если та же ошибка возникает снова, то, возможно, имеется аппаратная неисправность. В этом случае следует проинформировать сервисную службу Mitsubishi.
WDT ERROR (2) Ошибка WDT 2 (постоянный контроль)	25	стоп	Программа центрального процессора находится в бесконечном цикле (A3, A1N, A2N(S1), A3N, A3H).	Перевести центральный процессор в состояние "STOP" и выполнить сброс с помощью выключателя с ключом. Затем проверить программу на правильность программирования команд JMP, CJ и SCJ, а также указателя (P).
MAIN CPU DOWN (1) Неправильное функционирование главного центрального процессора (master) (постоянный контроль)	26	стоп	Неправильное функционирование или неисправность главного центрального процессора.	Обратиться в сервисную службу Mitsubishi.
UNIT VERIFY ERR. Ошибка коммуникации (постоянный контроль)	31	RUN (стоп)	Данные модуля ввода-вывода изменились после включения. Модуль ввода-вывода (включая специальный модуль) не находится в своем первоначальном слоте или вместо него вставлен другой модуль.	Для локализации модуля считать с помощью программатора специальные регистры с D9116 по D9123. Бит модуля, в котором возникла ошибка сверки, установлен на „1“. После восстановления первоначальной компоновки модулей необходимо выполнить сброс системы с помощью выключателя "RESET".
FUSE BREAK OFF Неисправен предохранитель (постоянный контроль)	32	RUN (стоп)	Неисправен предохранитель выходного модуля.	Для локализации модуля считать специальные регистры с D9100 по D9107 с помощью программатора. Бит модуля, предохранитель которого неисправен, установлен на „1“.
CONTROL BUS ERR. Ошибка в системной шине (проверка происходит при выполнении команд FROM/TO)	40	стоп	Выполнение команд FROM и/или TO не возможно. Во время передачи в специальный модуль возникла ошибка в системной шине.	Ошибка может быть вызвана неисправным специальным модулем, модулем центрального процессора или самой системной монтажной шиной. Поэтому следует тщательно проверить монтажную шину и модули и, если необходимо, обратиться в сервисную службу Mitsubishi.
SP.UNIT DOWN Неправильное функционирование специального модуля (проверка происходит при выполнении команд FROM/TO)	41	стоп	Доступ к специальному модулю после выполнения команды FROM и/или TO не сопровождался реакцией специального модуля. Запрашиваемый специальный модуль функционирует неправильно или неисправен.	Обратиться в сервисную службу Mitsubishi.

## Перечень кодов ошибок серии "A" (кроме AnA и AnAS)

### Перечень кодов ошибок серии "A" (кроме AnA и AnAS), продолжение

Сообщение об ошибке	Содержимое рег. D9008	Сост. центр. процессора	Причина	Устранение
LINK UNIT ERROR" Неправильно размещен интерфейсный модуль	42	стоп	Интерфейсный модуль для коммуникационной сети (например, AJ71R22, AJ71P22) находится на монтажной шине мастер-станции.	Вынуть модуль из монтажной шины мастер-станции, выполнить сброс системы с помощью выключателя "RESET" и повторить инициализацию.
I/O INT. ERROR" Ошибка прерывания	43	стоп	Было выполнено прерывание, хотя в системе нет модуля прерываний.	Обратиться в сервисную службу Mitsubishi.
SP.UNIT LAY ERROR" Сбой связи с модулем Computer-Link	44	стоп	В системе имеется более 2 модулей Computer-Link (например, AJ71C21 или AJ71C24), относящихся к одному центральному процессору. На монтажной шине имеется более одного интерфейсного модуля для коммуникационной сети (например, AJ71R22, AJ71P22). На монтажной шине имеется более одного модуля прерываний. Специальный модуль находится в слоте, которому в параметрах сопоставлен модуль ввода-вывода (или наоборот).	Удалить модули, которые имеются в системе или на монтажной шине в двойном экземпляре.  При неправильном присвоении адресов слота заново сконфигурировать их с помощью программатора.
SP. UNIT ERROR" Ошибочные настройки (проверка происходит при выполнении команды FROM/TO)	46	стоп	Команда FROM/TO обращается к станции в сети MELSECNET/MINI, которая не была определена при инициализации. Команда FROM/TO обращается к специальному модулю в сети MELSECNET, которого на установленном адресе не имеется.	Считать ошибочный шаг программы с помощью программатора и соответственно исправить строку программы с командой FROM/TO.
LINK PARA. ERROR" Ошибка в параметрах обмена данными	47	работа продолжается	В параметрах для установления диапазона передачи при обмене данными по коммуникационной сети имеются ошибки или эти параметры отсутствуют. Настройка общего количества локальных станций в сети равна „0“.	Проверить и откорректировать параметры. Если необходимо, ввести параметры заново. Дополнительно проверить настройку номеров станций.  Если эта ошибка возникает снова, то это означает, что имеется аппаратная неисправность. В этом случае следует проинформировать сервисную службу Mitsubishi.
OPERATION ERROR" Ошибка выполнения (проверка происходит при выполнении команды)	50	работа продолжается	Результат BCD-преобразования превышает допустимый диапазон (9999 или 99999999). Запрограммированный адрес операнда находится вне допустимого диапазона, поэтому выполнение операции невозможно. Программа содержит регистр файлов, для которого не указана область. Во время выполнения команды RTOP, RFRP, LWTP или LRDP возникла ошибка обработки. При использовании сети MELSECNET/MINI целевой номер станции в команде FROM/TO равен 0 или превышает 62.	Просмотреть шаги программы с помощью программатора и проверить их на возможные источники ошибок (диапазон адресов операндов, значения данных для BCD-преобразования, настройка параметра регистров файлов и т. п.).
MAIN CPU DOWN (2)" Неправильное функционирование главного центрального процессора (master) (ошибка при прерывании)	60	стоп	Команда INT обработана в области для программ микрокомпьютера. Неправильное функционирование центрального процессора из-за влияния напряжения помех (электронный шум и т. п.). Аппаратная неисправность	Стереть команду INT.  Если имеются сильные помехи, следует выяснить и удалить источник помех. Если устранить проблему не удается, следует заменить модуль центрального процессора.
BATTERY ERROR" Неисправность батареи (постоянный контроль)	70	RUN	Напряжение батареи снизилось ниже минимального предела. Батарея не подключена или подключена неправильно.	Проверить соединение батарей. Если необходимо, заменить батарею. Если используется память RAM или защита от исчезновения сетевого питания, то необходима батарея.

## 13.4 Перечень кодов ошибок серий AnA и AnAS

В следующей таблице дан обзор возможных кодов ошибок и соответствующих текстовых сообщений, а также указаны их возможные причины и способы устранения. Код ошибки записывается в специальный регистр D9008, подробный код – в специальный регистр D9091, а шаг программы, в котором возникла ошибка – в специальные регистры D9010 и D9011. В этой таблице перечислены лишь сообщения об ошибках, относящиеся к центральным процессорам AnA и AnAS.

Сообщение об ошибке	Содержимое рег. D9008	Содержимое рег. D9091	Причина	Устранение
INSTRCT CODE ERR“ Ошибочная команда (проверка происходит после переключения со STOP на RUN или выполнения команды)	10	101	Программа содержит команды, которые не входят в перечень команд центрального процессора и поэтому не могут в нем обрабатываться.	Считать с помощью программатора шаг программы, в котором возникла ошибка, и исправить соответствующее место в программе.  Если программа обрабатывается через запоминающее устройство ROM, следует проверить, подходит ли вставленный элемент ROM к данному центральному процессору.
		102	Индексация применена для 32-битной константы.	Считать с помощью программатора шаг программы, в котором возникла ошибка, и исправить соответствующее место в программе.
		103	Неправильно определен операнд, к которому обращается расширенная прикладная команда.	
		104	Расширенная прикладная команда имеет ошибочную структуру программирования.	
		105	Расширенная прикладная команда имеет ошибочную командную часть.	
		106	Индексация с помощью операндов Z или V находится в программе между командами LEDA/B IX и LEDA/B IXEND.	
		107	Индексация выполнена в отношении адресов операндов и заданных значений в выходном контакте таймера или счетчика. Индексация выполнена в отношении метки указателя, относящейся к цели перехода команды CJ, SCJ, CALL(P), JMP, LEDA/B FCALL или LEDA/B BREAK.	
		108	Ошибка имеет иную причину кроме названных в кодах с 101 по 107.	

Перечень кодов ошибок процессоров AnA и AnAS (продолжение)

Сообщение об ошибке	Содержимое рег. D9008	Содержимое рег. D9091	Причина	Устранение
<p>PARAMETER ERROR“</p> <p>Ошибочный параметр (проверка происходит при включении, сбросе и переключении STOP/RUN или PAUSE/RUN)</p>	11	111	Для одной или нескольких из нижеуказанных позиций указанные в параметрах области памяти находятся вне допустимого диапазона центрального процессора:  основная программа и подпрограмма, программа микрокомпьютера, комментарии к регистрам файлов, фиксация состояния, выборочная трассировка и расширенные регистры файлов.	Считать параметры из памяти центрального процессора и проверить их содержимое. Изменить ошибочные значения параметров и снова записать исправленные значения в память.
		112	Общий объем памяти для основной программы и подпрограммы, программы микрокомпьютера, комментариев к регистрам файлов, фиксации состояния, выборочной трассировки и расширенных регистров файлов превышает максимальную емкость кассеты памяти.	
		113	Ошибочно определена область промежуточной памяти (для фиксации состояния), установленная с помощью параметров или операндов M, L или S.	
		114	Ошибка обработки (Sum check).	
		115	В параметрах неправильно определена одна из следующих позиций:  Дистанционный контакт RUN/PAUSE, режим при возникновении неисправности, вид индикации в сигнализаторах ошибок, вид индикации STOP/RUN.	
		116	Неправильная настройка автоматического обновления сетевых данных в параметрах.	
		117	Неправильная установка таймера в параметрах.	
		118	Неправильная установка счетчика в параметрах.	
<p>MISSING END INS.“</p> <p>Отсутствует команда END (проверка происходит после переключения STOP/RUN)</p>	12	121	В основной программе нет команды END/FEND.	Вставить команду END в конце программы/подпрограммы.
		122	В подпрограмме нет команды END/FEND (если подпрограмма установлена с помощью параметра).	
<p>CAN'T EXECUTE (P)“</p> <p>Невыполнимый переход программы (проверка происходит при выполнении команды)</p>	13	131	Для обозначения цели перехода двум или более местам в программе присвоены идентичные адреса указателей (P) или указателей прерывания (I).	Удалить идентичные адреса указателей в целях переходов.
		132	В программе нет цели перехода, к которой обращается команда CJ, SCJ, CALL(P), JMP, LEDA/B FCALL или LEDA/B BREAK, или эта цель расположена после команды END.	Считать шаг программы, в котором возникла ошибка, и вставить адрес перехода (P) в правильном месте программы.

## Перечень кодов ошибок процессоров AnA и AnAS (продолжение)

Сообщение об ошибке	Содержимое рег. D9008	Содержимое рег. D9091	Причина	Устранение
CAN'T EXECUTE (P) Невыполнимый переход программы (проверка происходит при выполнении команды)	13	133	Выполнена команда RET, хотя в программе нет команды CALL. В программе имеется команда NEXT и LEDA/B BREAK, хотя нет соответствующей команды FOR. Команды CALL(P) или FOR вложены глубже 5 уровней. После команды CALL(P) или FOR отсутствует соответствующая команда RET или NEXT.	Считать с помощью программатора шаг программы, в котором возникла ошибка, и исправить соответствующее место в программе.  Уменьшить количество уровней вложения в месте применения команд CALL(P)-RET и FOR-NEXT – это количество должно быть не больше 5.
		134	В программе имеется команда CHG, однако нет подпрограммы.	Считать номер ошибочного шага программы и стереть команду CHG.
		135	Команды LEDA/B IX и LEDA IXEND запрограммированы не парами. Программа содержит более 32 пар команд LEDA/B IX и LEDA IXEND.	Считать номер ошибочного шага программы и исправить соответствующую строку программы.
CHK FORMAT ERR Ошибка в формате CHK (проверка происходит после переключения STOP/RUN или PAUSE/RUN)	14	141	В блоке команд CHK (включая NOP) находятся иные команды кроме LDX, LDIX, ANDX и ANIX.	Проверить блок команд CHK в программе на возможные причины ошибки на основе более подробных кодов ошибок, указанных рядом. Исправить ошибки.
		142	Команда CHK вставлена в программу более одного раза.	
		143	Блок CHK содержит более 150 входных контактов (контрольных условий).	
		144	Команда LEDA/CHK запрограммирована не в паре с командой LEDA/CHKEND или запрограммировано более 1 пары этих команд.	
		145	Над блоком команд CHK нет команды CJ с входным условием.	
		146	Адрес операнда D1 в команде CHK D1 D2 не соответствует адресу операнда над командой CJ.	
		147	В блоке команд CHK осуществляется индексация.	
148	Программа содержит более одной пары команд LEDA/CHK и LEDA/CHKEND. В блоке команд LEDA/CHK и LEDA/CHKEND имеется более 6 входных контактов (контрольных условий). Контрольные условия в блоке команд LEDA/CHK и LEDA/CHKEND не содержат входного или выходного операнда (X и Y). Блок команд LEDA/CHK и LEDA/CHKEND охватывает более 256 шагов программы.	Проверить блок команд CHK в программе на возможные причины ошибки на основе более подробных кодов ошибок, указанных рядом. Исправить ошибки.		



Перечень кодов ошибок процессоров AnA и AnAS (продолжение)

Сообщение об ошибке	Содержимое рег. D9008	Содержимое рег. D9091	Причина	Устранение
CAN'T EXECUTE (1) Выполнение не возможно (проверка происходит при выполнении прерывания)	15	151	Команда IRET находится не в программе прерывания, а в иной части программы.	Проверить, не находится ли команда IRET вне программы прерывания, и стереть соответствующее место в программе.
		152	В программа прерывания нет команды IRET.	Исследовать программу прерывания на отсутствие команды IRET и вставить эту команду.
		153	Адрес указателя прерывания (I), по которому происходит обращение к модулю прерываний, ошибочен или отсутствует. Ошибочный адрес указателя записывается в специальный регистр D9011.	Считать содержимое специального регистра D9011 и проверить наличие программы прерывания, а также ее сопоставление модулю прерываний. Если необходимо, стереть указатель прерывания, запрограммированный дважды.
CASSETTE ERROR Доступ к кассете памяти не возможен	16	—	В модуле центрального процессора нет кассеты памяти или кассета неисправна.	Вставить или заменить кассету памяти (при отключенном напряжении питания).
RAM ERROR Ошибка при обращении к памяти RAM (проверка происходит после включения)	20	201	При обращении к сохраненной основной программе возникла ошибка.	Обратиться в сервисную службу Mitsubishi.
		202	При обращении к оперативной памяти центрального процессора возникла ошибка.	
		203	При обращении к памяти операндов центрального процессора возникла ошибка.	
		204	При обращении к памяти адресов центрального процессора возникла ошибка.	
OPE. CIRCUIT ERR Неисправность в рабочем контуре (проверка происходит после включения и сброса)	21	211	Рабочий контур, отвечающий за обработку индексов в центральном процессоре, работает неправильно.	Обратиться в сервисную службу Mitsubishi.
		212	Аппаратная логика центрального процессора работает неправильно.	
		213	Неправильно работает контур, отвечающий за обработку программы в центральном процессоре.	
WDT ERROR Ошибка WDT (проверка происходит после выполнения команды END)	22	—	Время цикла программы сильно увеличилось. Время цикла увеличилось из-за кратковременного падения напряжения, произошедшего во время прогона программы.	Проверить и заново рассчитать время цикла программы. Время цикла программы можно уменьшить с помощью команд CJ и т. п. Если в структуре программы ошибок нет, считать содержимое специального регистра D9005. Если его значение не равно "0", то это указывает на недостаточное напряжение питания. В этом случае проверить электропитание и устранить причину потери напряжения.
END NOT EXECUTE Не удалось выполнить команду END (проверка происходит во время выполнения команды END)	24	241	Программа отработана без выполнения команды END. Во время выполнения команды END процессор из-за помех (наведенных напряжений и т. п.) считал другой код команды. Под действием помех команда END преобразовалась в другую команду.	Выполнить сброс и заново перевести центральный процессор в режим RUN. Если та же ошибка возникает снова, то, возможно, имеется аппаратная неисправность. В этом случае следует проинформировать сервисную службу Mitsubishi.

## Перечень кодов ошибок процессоров AnA и AnAS (продолжение)

Сообщение об ошибке	Содержимое рег. D9008	Содержимое рег. D9091	Причина	Устранение
MAIN CPU DOWN (1) Неправильное функционирование главного центрального процессора (master) (постоянный контроль)	26	—	Неправильное функционирование или неисправность главного центрального процессора.	Обратиться в сервисную службу Mitsubishi.
UNIT VERIFY ERR. Ошибка коммуникации (постоянный контроль)	31	—	После включения электропитания изменились данные модуля ввода-вывода.  Модуль ввода-вывода (включая специальный модуль) не находится в своем первоначальном слоте или вместо него вставлен другой модуль.	Для определения ошибочного модуля считать специальные регистры с D9116 по D9123 с помощью программатора. Бит модуля, в отношении которого возникла ошибка сравнения, установлен на „1“.  После восстановления первоначальной компоновки модулей необходимо выполнить сброс системы с помощью выключателя "RESET".
„FUZE BREAK OFF“ Неисправен предохранитель (постоянный контроль)	32	—	Неисправен предохранитель выходного модуля.	Для локализации модуля считать специальные регистры с D9100 по D9107 с помощью программатора. Бит модуля, в котором неисправен предохранитель, установлен на „1“.
CONTROL BUS ERR. Неисправность в системной шине	40	401 402	Выполнение команд FROM и/или TO не возможно. Во время передачи в специальный модуль возникла ошибка в системной шине.  При присвоении адресов с помощью параметров доступ к специальному модулю в начале коммуникации не возможен. После возникновения ошибки начальный адрес специального модуля сохраняется в специальном регистре D9011.	Ошибка может быть вызвана неисправным специальным модулем, модулем центрального процессора или самой системной монтажной шиной. Поэтому следует тщательно проверить монтажную шину и модули и, если необходимо, обратиться в сервисную службу Mitsubishi.
SP.UNIT DOWN Неправильное функционирование специального модуля	41	411 412	Доступ к специальному модулю после выполнения команды FROM и/или TO не сопровождался реакцией специального модуля.  При присвоении адресов с помощью параметров доступ к специальному модулю в начале коммуникации не возможен. После возникновения ошибки начальный адрес специального модуля сохраняется в специальном регистре D9011.	Обратиться в сервисную службу Mitsubishi.
LINK UNIT ERROR Интерфейсный модуль неправильно размещен	42	—	Интерфейсный модуль для коммуникационной сети (например, AJ71R22, AJ71P22) вставлен в монтажную шину мастер-станции.	Вынуть модуль из монтажной шины мастер-станции, выполнить сброс системы с помощью выключателя "RESET" и повторить инициализацию.
I/O INT. ERROR Ошибка прерывания	43	—	Было выполнено прерывание, хотя в системе нет модуля прерываний.	Обратиться в сервисную службу Mitsubishi.

Перечень кодов ошибок процессоров AnA и AnAS (продолжение)

Сообщение об ошибке	Содержимое рег. D9008	Содержимое рег. D9091	Причина	Устранение
SP.UNIT LAY ERROR" Ошибка, связанная с модулем Computer-Link	44	441	В параметрах неправильно присвоены адреса. Специальному модулю присвоен адрес модуля ввода-вывода (или наоборот).	При неправильном присвоении адресов специального модуля их следует заново сконфигурировать с помощью программатора.
		442	В системе имеется более 8 специальных модулей (кроме A161-S1), способных выполнять прерывание для центрального процессора.	Уменьшить до 8 количество модулей, способных выполнять прерывание.
		443	На монтажной шине имеется более одного интерфейсного модуля для коммуникационной сети (например, AJ71R22, AJ71P22).	Уменьшить количество модулей AJ71R22, AJ71P22 до „1“.
		444	На монтажной шине модуля центрального процессора находятся более 6 модулей Computer-Link.	Ограничить количество модулей Computer-Link шестью.
		445	На монтажной шине имеется более одного модуля A161-S1.	Уменьшить количество модулей A161-S1 до одного.
		446	Параметры автоматического обновления MNTMINI для некоторых модулей сети не соответствуют действительной ситуации.	Заново присвоить параметры автоматического обновления MNTMINI.
		447	Превышено максимальное количество специальных модулей, сопоставленных модулю центрального процессора и способных обрабатывать расширенный перечень команд (максимальное количество не должно превышать 1344).	Уменьшить общее количество специальных модулей в системе.
SP. UNIT ERROR" Ошибочные настройки (проверка происходит при выполнении команды FROM/TO)	46	461	Модуль, к которому обращается команда FROM/TO, не является специальным модулем.	Считать шаг программы, в котором возникла ошибка, с помощью программатора и исправить содержимое команды FROM/TO.
		462	Модуль, к которому обращается расширенная прикладная команда процессора AnA или AnU, не является специальным модулем или является неправильным специальным модулем.	Считать шаг программы, в котором возникла ошибка, с помощью программатора и исправить содержимое соответствующей команды.
LINK PARA. ERROR" Ошибка в параметрах обмена данными	47	—	В параметрах для установления диапазона передачи при обмене данными по коммуникационной сети имеются ошибки или эти параметры отсутствуют. Настройка общего количества локальных станций, имеющихся в сети, равна „0“.	Проверить и откорректировать параметры. Если необходимо, ввести параметры заново. Дополнительно проверить настройку номеров станций. Если эта ошибка возникает снова, то это означает, что имеется аппаратная неисправность. В этом случае следует проинформировать сервисную службу Mitsubishi.
BATTERY ERROR" Неисправность батареи (постоянный контроль)	70	—	Напряжение батареи снизилось ниже минимального предела. Батарея не подключена или подключена неправильно.	Проверить соединение батареи. Если необходимо, заменить батарею. Если используется память RAM или защита от исчезновения сетевого питания, то необходима батарея.

Перечень кодов ошибок центральных процессоров AnA и AnAS (продолжение)

Сообщение об ошибке	Содержимое рег. D9008	Содержимое рег. D9091	Причина	Устранение
OPERATION ERROR" Ошибка выполнения (проверка происходит при выполнении команды)	50	501	Требуется обработать регистры файлов (R) вне установленных областей адресов операндов и адресов блоков регистров файлов. В программе требуется обработать регистр файлов (R), хотя объем регистров не был задан.	С помощью программатора считать шаг программы, в котором возникла ошибка, и откорректировать соответствующую ошибочную команду.
		502	В команде запрограммировано недопустимое сочетание операндов.	
		503	Сохраненное значение данных или константа заданного операнда находится вне максимально допустимого диапазона.	
		504	Заданное количество данных, которое требуется обработать, превышает максимально возможный диапазон.	
		505	Адрес станции, к которому обращается команда LEDA/B LRDP, LCDA/B LWTP, LRDP или LWTP, не соответствует локальной станции. По адресу ввода-вывода, к которому обращается команда LEDA/B RFRP, LEDA/B RTOP, RFRP или RTOP, не находится удаленная станция.	
		506	По адресу ввода-вывода, к которому обращается команда LEDA/B RFRP, LEDA/B RTOP, RFRP или RTOP, не находится специальный модуль.	
		507	Во время обработки команд в режиме "divided processing mode" к модулю AD57(S1) или AD58 было выполнено обращение через сеть. Во время обработки команд в режиме "divided processing mode" модуля AD57(S1) или AD58 было одновременно осуществлено обращение еще к одному модулю AD57(S1) или AD58.	С помощью программатора считать шаг программы, в котором возникла ошибка, и откорректировать соответствующую ошибочную команду.
509	Через сеть поступила команда, которую модуль дистанционного терминала не может обработать. При поступлении команды PRC на модуль дистанционного терминала область памяти для регистрации коммуникационного запроса занята. Команда PIDCONT выполнена без соответствующей команды PIDINIT. Команда PID57 выполнена без соответствующей команды PIDINIT или PIDCONT.	Считать номер ошибочного шага программы и исправить соответствующую строку программы с учетом данных модуля.  С помощью специального маркера M9081 или регистра данных D9081 предусмотрите блокировку доступа к модулю дистанционного терминала во время выполнения команды PRC.  После выполнения команды PIDINIT должна быть запрограммирована команда PIDCONT. Выполнение команды PID57 допускается лишь после выполнения команд PIDINIT и PIDCONT.		
MAIN CPU DOWN (2)" Неправильное функционирование главного центрального процессора (master) (ошибка при прерывании)	60	—	Команда INT обработана в области для программ микрокомпьютера. Неправильное функционирование центрального процессора под влиянием напряжения помех. Аппаратная неисправность	Стереть команду INT.  Если имеются сильные помехи, следует выяснить и удалить источник помех. Если устранить проблему не удастся, следует заменить модуль центрального процессора.

# А Приложение А

## А.1 Определение времени обработки

Время обработки – это сумма:

- времени обработки каждой из команд,
- времени обработки команды END. Это время состоит из времени, необходимого для выполнения команды END, времени обновления сети MELSECNET, если таковая подключена, времени, необходимого для обмена данными с периферийными устройствами, и времени, затрачиваемого на последовательную коммуникацию,
- времени обновления входов и выходов, рассчитываемого следующим образом:

$$\text{Время обновл. входов и выходов} = \frac{\text{сумма входов}}{16} \times N1 + \frac{\text{сумма выходов}}{16} \times N2$$

В нижеследующей таблице указано время N1 и N2 для центральных процессоров QnA и MELSEC System Q.

Тип центрального процессора	N1 (мкс)			N2 (мкс)		
	Главная монтажная шина Q	Расширительная монтажная шина Q	Расширительная монтажная шина QnA	Главная монтажная шина Q	Расширительная монтажная шина Q	Расширительная монтажная шина QnA
Q00JCPU	2.5	3.3	—	1.3	2.3	—
Q00CPU	2.4	3.2	—		2.3	—
Q01CPU	2.3	3.1	—		2.3	—
Q02CPU	2.2	2.9	4.3		2.1	3.5
Q02HCPU Q06HCPU Q12HCPU Q12PHCPU Q25HCPU Q25PHCPU	1.7	2.4	3.7		2.1	3.5
Q2ASCPU (S1) Q2ACPU	5.2			5.0		
Q3ACPU	4.8			4.65		
Q2ASHCPU (S1) Q4ACPU Q4ARCPU	4.34			4.26		

## А.2 Значения времени обработки

На следующих страницах в табличном виде указано время обработки всех команд. Время обработки зависит от значений в исходных и целевых данных. Указанные в таблицах значения служат в качестве ориентировочных данных для расчета общего времени обработки.

Время обработки команды не включает в себя время индексации. Если команда не обрабатывается, время обработки можно рассчитать следующим образом:

Тип центрального процессора	Время обработки при не выполненной команде (мкс)
Q00JCPU	0.20 x (количество шагов на команду +1)
Q00CPU	0.16 x (количество шагов на команду +1)
Q01CPU	0.10 x (количество шагов на команду +1)
Q02CPU)	0.079 x (количество шагов на команду +1)
Q02HCPU Q06HCPU Q12HCPU Q12PHCPU Q25HCPU Q25PHCPU	0.034 x (количество шагов на команду +1)
Q2ASCPU (S1) Q2ACPU	0.20 x (количество шагов на команду +1)
Q3ACPU	0.15 x (количество шагов на команду +1)
Q2ASHCPU (S1) Q4ACPU Q4ARCPU	0.075 x (количество шагов на команду +1)

**А.2.1 Перечень значений времени обработки (серии QnA и System Q)**

Команда	Обработка (операнд)		Время обработки (мкс)									
			Серия QnA				System Q					
			Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H	
LD	например, X0		0.20	0.15	0.075	0.075	0.20	0.16	0.10	0.079	0.034	
LDI												
AND												
ANI												
OR												
ORI												
LDP	например D0.0		6.6	5.0	2.5	2.5	0.30	0.24	0.15	0.158	0.068	
LDF												
ANDP												
ANDF												
ORP												
ORF												
ANB			0.20	0.15	0.075	0.075	0.20	0.16	0.10	0.079	0.034	
ORB												
MPS												
MRD												
MPP												
INV												не выполнена
	выполнена											
MEP	не выполнена	2.0	1.5	0.75	0.75	0.30	0.24	0.15	0.173	0.073		
MEF	выполнена											
EGP	неизменное состояние	0.6	0.3	0.15	0.15	0.20	0.16	0.10	0.158	0.068		
	изменяющееся состояние (выкл./вкл. или вкл./выкл.)					17	9.5	9.4				
EGF	неизменное состояние	0.6	0.3	0.15	0.15	17	9.5	9.4	0.158	0.068		
	изменяющееся состояние (выкл./вкл. или вкл./выкл.)					18	14	14				
OUT	кроме F, T, C	неизменное состояние	0.40	0.30	0.15	0.15	0.20	0.16	0.10	0.158	0.068	
		меняющееся состояние (выкл./вкл. или вкл./выкл.)										
	D0.0	неизменное состояние	0.40	0.30	0.15	0.15	0.40	0.32	0.20	0.158	0.068	
		меняющееся состояние (выкл./вкл. или вкл./выкл.)										
	F	не выполнена	7.0	5.3	2.7	2.7	24	20	19	2.8	1.2	
		выполнена	индикация	167	126	63	63	260	210	200	162	69.7
			индикация окончена	166	125	62	62	205	165	155	126	54
	T	не выполнена	1.6	1.2	0.6	0.6	1.1	0.88	0.55	0.63	0.27	
		выполнена										после истечения
												дополнительно с
	C	не выполнена	1.6	1.2	0.6	0.6	1.1	0.88	0.55	0.63	0.27	
		выполнена										после истечения
дополнительно с												K
OUTH	не выполнена	1.6	1.2	0.6	0.6	1.1	0.88	0.55	0.63	0.27		
	выполнена										после истечения	
											дополнительно с	K
SET	все кроме F и D0.0	не выполнена	0.40	0.30	0.15	0.15	0.20	0.16	0.10	0.158	0.068	
		выполнена										неизменное состояние
	D0.0	не выполнена	0.40	0.30	0.15	0.15	0.40	0.32	0.20	0.158	0.068	
		выполнена										неизменное состояние
	F	не выполнена	1.2	0.90	0.45	0.45	0.50	0.44	0.25	0.47	0.20	
		выполнена	индикация	277	208	104	104	255	205	195	161	69
	индикация окончена		1.2	0.90	0.45	0.45	195	160	150	0.47	0.20	

Команда	Обработка (операнд)		Время обработки (мкс)									
			Серия QnA				System Q					
			Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H	
RST	все кроме нижнеза- ванных операндов	не выполнена	0.40	0.30	0.15	0.15	0.20	0.16	0.10	0.158	0.068	
		выполнена										неизменное состояние
		выполнена										меняющееся состояние
	D0.0	выполнена	не выполнена	0.40	0.30	0.15	0.15	0.40	0.32	0.20	0.158	0.068
			неизменное состояние									
			меняющееся состояние									
	SM		не выполнена	0.40	0.30	0.15	0.15	0.20	0.16	0.10	0.158	0.068
			выполнена									
	F	выполнена	не выполнена	1.2	0.90	0.45	0.45	0.48	0.44	0.25	0.47	0.20
			индикация	148	112	56	56	75	69	65	90	38
			индикация окончена	1.2	0.90	0.45	0.45	43	35	33	0.47	0.20
	T, C		не выполнена	1.4	1.1	0.6	0.6	0.80	0.64	0.40	0.63	0.27
			выполнена					1.0	0.80	0.50		
	D		не выполнена	0.60	0.45	0.23	0.23	0.40	0.32	0.20	0.24	0.10
			выполнена					0.60	0.48	0.30		
	Z		не выполнена	1.2	0.90	0.45	0.45	0.50	0.40	0.25	0.47	0.20
			выполнена	10.8	8.1	4.1	4.1	9.4	7.9	7.4	4.3	1.9
	R		не выполнена	1.0	0.75	0.38	0.38	—	0.32	0.20	0.40	0.17
выполнена			—					0.48	0.30			
PLS			2.6	2.0	0.98	0.98	12	9.5	9.2	1.0	0.44	
PLF			2.6	2.0	0.98	0.98	11	9.5	8.9	1.0	0.44	
FF	Y	не выполнена	1.2	0.90	0.45	0.45	0.68	0.40	0.25	0.47	0.20	
		выполнена					7.5	6.2	5.7			
DELTA	DY0	не выполнена	1.2	0.90	0.45	0.45	0.50	0.40	0.25	0.47	0.20	
		выполнена	16.8	14.1	11.1	11.1	26	21	21	5.9	2.6	
DELTAP	DY0	не выполнена	1.2	0.90	0.45	0.45	0.48	0.40	0.25	0.47	0.20	
		выполнена	16.8	14.1	11.1	11.1	58	45	43	5.9	2.6	
SFT SFTP		не выполнена	1.2	0.90	0.45	0.45	0.50	0.34	0.25	0.47	0.20	
		выполнена	4.2	3.2	1.6	1.6	12	8.7	8.3	1.66	0.71	
MC		M0.0	0.60	0.45	0.23	0.23	0.40	0.32	0.20	0.24	0.10	
		D.0					3.3	2.9	2.8			
MCR			0.20	0.15	0.075	0.075	0.20	0.16	0.10	0.079	0.034	
FEND END		контроль на ошибки выполнен	1643	1236	618	618	660	530	480	348	150	
		без контроля на ошибки: — проверка батареи — проверка плавкого предохранителя — проверка модуля ввода-вывода	1106	832	416	416	660	530	480	359	150	
NOP			0.2	0.15	0.075	0.075	0.20	0.16	0.10	0.079	0.034	
NOPLF PAGE			0.2	0.15	0.075	0.075	0.20	0.16	0.10	0.79	0.034	
LD=		с непрерывностью	3.8	2.9	1.5	1.5	0.80	0.64	0.40	0.24	0.10	
		без непрерывности	3.6	2.7	1.4	1.4						
AND=	выполнена	не выполнена	1.4	1.1	0.55	0.55	0.70	0.56	0.35	0.24	0.10	
		с непрерывностью	2.8	2.1	1.1	1.1	0.80	0.64	0.40			
		без непрерывности	3.2	2.4	1.2	1.2						
OR=	выполнена	не выполнена	1.4	1.1	0.55	0.55	0.70	0.56	0.35	0.24	0.10	
		с непрерывностью	3.8	2.9	1.5	1.5	0.80	0.64	0.40			
		без непрерывности	2.8	2.1	1.1	1.1						
LD<>		с непрерывностью	4.4	3.3	1.7	1.7	0.80	0.64	0.40	0.24	0.10	
		без непрерывности	3.6	2.7	1.4	1.4						
AND<>	выполнена	не выполнена	1.4	1.1	0.55	0.55	0.70	0.56	0.35	0.24	0.10	
		с непрерывностью	2.8	2.1	1.1	1.1	0.80	0.64	0.40			
		без непрерывности	3.2	2.4	1.2	1.2						



Команда	Обработка (операнд)		Время обработки (мкс)								
			Серия QnA				System Q				
			Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
OR<>	не выполнена		1.4	1.1	0.55	0.55	0.70	0.56	0.35	0.24	0.10
	выполнена	с непрерывностью	3.8	2.9	1.5	1.5	0.80	0.64	0.40		
		без непрерывности	2.8	2.1	1.1	1.1					
LD>	с непрерывностью		4.4	3.3	1.7	1.7	0.80	0.64	0.40	0.24	0.10
	без непрерывности		3.6	2.7	1.4	1.4					
AND>	не выполнена		1.4	1.1	0.55	0.55	0.70	0.56	0.35	0.24	0.10
	выполнена	с непрерывностью	2.8	2.1	1.1	1.1	0.80	0.64	0.40		
		без непрерывности	3.2	2.4	1.2	1.2					
OR>	не выполнена		1.4	1.1	0.55	0.55	0.70	0.56	0.35	0.24	0.10
	выполнена	с непрерывностью	3.8	2.9	1.5	1.5	0.80	0.64	0.40		
		без непрерывности	2.8	2.1	1.1	1.1					
LD<=	с непрерывностью		4.4	3.3	1.7	1.7	0.80	0.64	0.40	0.24	0.10
	без непрерывности		3.6	2.7	1.4	1.4					
AND<=	не выполнена		1.4	1.1	0.55	0.55	0.70	0.56	0.35	0.24	0.10
	выполнена	с непрерывностью	2.8	2.1	1.1	1.1	0.80	0.64	0.40		
		без непрерывности	3.2	2.4	1.2	1.2					
OR<=	не выполнена		1.4	1.1	0.55	0.55	0.70	0.56	0.35	0.24	0.10
	выполнена	с непрерывностью	3.8	2.9	1.5	1.5	0.80	0.64	0.40		
		без непрерывности	2.8	2.1	1.1	1.1					
LD<	с непрерывностью		4.4	3.3	1.7	1.7	0.80	0.64	0.40	0.24	0.10
	без непрерывности		3.6	2.7	1.4	1.4					
AND<	не выполнена		1.4	1.1	0.55	0.55	0.70	0.56	0.35	0.24	0.10
	выполнена	с непрерывностью	2.8	2.1	1.1	1.1	0.80	0.64	0.40		
		без непрерывности	3.2	2.4	1.2	1.2					
OR<	не выполнена		1.4	1.1	0.55	0.55	0.70	0.56	0.35	0.24	0.10
	выполнена	с непрерывностью	3.8	2.9	1.5	1.5	0.80	0.64	0.40		
		без непрерывности	2.8	2.1	1.1	1.1					
LD>=	с непрерывностью		4.4	3.3	1.7	1.7	0.80	0.64	0.40	0.24	0.10
	без непрерывности		3.6	2.7	1.4	1.4					
AND>=	не выполнена		1.4	1.1	0.55	0.55	0.70	0.56	0.35	0.24	0.10
	выполнена	с непрерывностью	2.8	2.1	1.1	1.1	0.80	0.64	0.40		
		без непрерывности	3.2	2.4	1.2	1.2					
OR>=	не выполнена		1.4	1.1	0.55	0.55	0.70	0.56	0.35	0.24	0.10
	выполнена	с непрерывностью	3.8	2.9	1.5	1.5	0.80	0.64	0.40		
		без непрерывности	2.8	2.1	1.1	1.1					
LDD=	с непрерывностью		5.0	3.8	1.9	1.9	1.0	0.80	0.50	0.55	0.24
	без непрерывности		4.2	3.2	1.6	1.6				0.39	0.17
ANDD=	не выполнена		1.4	1.1	0.55	0.55	0.80	0.64	0.40	0.39	0.17
	выполнена	с непрерывностью	3.4	2.6	1.3	1.3	1.0	0.80	0.50	0.55	0.24
		без непрерывности	3.8	2.9	1.5	1.5				0.39	0.17
ORD=	не выполнен		1.4	1.1	0.55	0.55	0.80	0.64	0.40	0.39	0.17
	с	с непрерывностью	4.4	3.3	1.7	1.7	1.0	0.80	0.50	0.55	0.24
		без непрерывности	3.4	2.6	1.3	1.3					
LDD<>	с непрерывностью		5.0	3.8	1.9	1.9	1.0	0.80	0.50	0.55	0.24
	без непрерывности		4.2	3.2	1.6	1.6					
ANDD<>	не выполнена		1.4	1.1	0.55	0.55	0.80	0.64	0.40	0.39	0.17
	выполнена	с непрерывностью	3.4	2.6	1.3	1.3	1.0	0.80	0.50	0.55	0.24
		без непрерывности	3.8	2.9	1.5	1.5					
ORD<>	не выполнена		1.4	1.1	0.55	0.55	0.80	0.64	0.40	0.39	0.17
	выполнена	с непрерывностью	4.4	3.3	1.7	1.7	1.0	0.80	0.50	0.55	0.24
		без непрерывности	3.4	2.6	1.3	1.3					
LDD>	с непрерывностью		3.8	2.9	1.5	1.5	1.0	0.80	0.50	0.55	0.24
	без непрерывности		4.2	3.2	1.6	1.6					

Команда	Обработка (операнд)		Время обработки (мкс)									
			Серия QnA				System Q					
			Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H	
ANDD>	не выполнена		1.4	1.1	0.55	0.55	0.80	0.64	0.40	0.39	0.17	
	выполнена	с непрерывностью	2.8	2.1	1.1	1.1	1.0	0.80	0.50	0.55	0.24	
		без непрерывности	3.8	2.9	1.5	1.5						
ORD>	не выполнена		1.4	1.1	0.55	0.55	0.80	0.64	0.40	0.39	0.17	
	выполнена	с непрерывностью	3.8	2.9	1.5	1.5	1.0	0.80	0.50	0.55	0.24	
		без непрерывности	3.4	2.6	1.3	1.3						
LDD<=	с непрерывностью		4.4	3.3	1.7	1.7	1.0	0.80	0.50	0.55	0.24	
	без непрерывности		3.6	2.7	1.4	1.4						
ANDD<=	не выполнена		1.4	1.1	0.55	0.55	0.80	0.64	0.40	0.39	0.17	
	выполнена	с непрерывностью	3.4	2.6	1.3	1.3	1.0	0.80	0.50	0.55	0.24	
		без непрерывности	3.2	2.4	1.2	1.2						
ORD<=	не выполнена		1.4	1.1	0.55	0.55	0.80	0.64	0.40	0.39	0.17	
	выполнена	с непрерывностью	4.4	3.3	1.7	1.7	1.0	0.80	0.50	0.55	0.24	
		без непрерывности	2.8	2.1	1.1	1.1						
LDD<	с непрерывностью		3.8	2.9	1.5	1.5	1.0	0.80	0.50	0.55	0.24	
	без непрерывности		4.2	3.2	1.6	1.6						
ANDD<	не выполнена		1.4	1.1	0.55	0.55	0.80	0.64	0.40	0.39	0.17	
	выполнена	с непрерывностью	2.8	2.1	1.1	1.1	1.0	0.80	0.50	0.55	0.24	
		без непрерывности	3.8	2.9	1.5	1.5						
ORD<	не выполнена		1.4	1.1	0.55	0.55	0.80	0.64	0.40	0.39	0.17	
	выполнена	с непрерывностью	3.8	2.9	1.5	1.5	1.0	0.80	0.50	0.55	0.24	
		без непрерывности	3.4	2.6	1.3	1.3						
LDD>=	с непрерывностью		4.4	3.3	1.7	1.7	1.0	0.80	0.50	0.55	0.24	
	без непрерывности		3.6	2.7	1.4	1.4						
ANDD>=	не выполнена		1.4	1.1	0.55	0.55	0.80	0.64	0.40	0.39	0.17	
	выполнена	с непрерывностью	3.4	2.6	1.3	1.3	1.0	0.80	0.50	0.55	0.24	
		без непрерывности	3.2	2.4	1.2	1.2						
ORD>=	не выполнена		1.4	1.1	0.55	0.55	0.80	0.64	0.40	0.39	0.17	
	выполнена	с непрерывностью	4.4	3.3	1.7	1.7	1.0	0.80	0.50	0.55	0.24	
		без непрерывности	2.8	2.1	1.1	1.1						
LDE=	обычная точность	с непрерывностью	235	177	89	35	—	—	—	93	40	
		без непрерывности	231	174	87	87	—	—	—	92		
	двойная точность	с непрерывностью	—	—	—	—	—	—	—	93	40	
		без непрерывности	—	—	—	—	—	—	—	92		
ANDE=	обычная точность	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.55	0.24	
		выполнена	с непрерывн.	234	176	88	35	—	—	—		93
			без непрерывности	230	172	86	86	—	—	—		92
	двойная точность	не выполнена	—	—	—	—	—	—	—	—	—	
		выполнена	с непрерывн.	—	—	—	—	—	—	—		93
			без непрерывности	—	—	—	—	—	—	—		92
ORE=	обычная точность	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.55	0.24	
		выполнена	с непрерывн.	234	176	88	35	—	—	—		93
			без непрерывности	230	172	86	86	—	—	—		92
	двойная точность	не выполнена	—	—	—	—	—	—	—	—	—	
		выполнена	с непрерывн.	—	—	—	—	—	—	—		93
			без непрерывности	—	—	—	—	—	—	—		92
LDE<>	обычная точность	с непрерывностью	231	174	87	35	—	—	—	92	40	
		без непрерывности	234	176	88	88	—	—	—			
	двойная точность	с непрерывностью	—	—	—	—	—	—	—	92	40	
		без непрерывности	—	—	—	—	—	—	—			

Команда	Обработка (операнд)		Время обработки (мкс)									
			Серия QnA				System Q					
			Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H	
ANDE<>	обычная точность	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.55	0.24	
		выполнена	с непрерывн.	230	172	86	34	—	—	—	92	40
			без непрерывности	234	176	88	88	—	—	—	93	
	двойная точность	не выполнена	—	—	—	—	—	—	—	—	—	
		выполнена	с непрерывн.	—	—	—	—	—	—	—	92	40
			без непрерывности	—	—	—	—	—	—	—		
ORE<>	обычная точность	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.55	0.24	
		выполнена	с непрерывн.	231	174	87	35	—	—	—	93	40
			без непрерывности	234	176	88	88	—	—	—	92	
	двойная точность	не выполнена	—	—	—	—	—	—	—	—	—	
		выполнена	с непрерывн.	—	—	—	—	—	—	—	93	40
			без непрерывности	—	—	—	—	—	—	—	92	
LDE>	обычная точность	с непрерывностью	231	174	87	35	—	—	—	92	40	
		без непрерывности	234	176	88	88	—	—	—			
	двойная точность	с непрерывностью	—	—	—	—	—	—	—	92	40	
		без непрерывности	—	—	—	—	—	—	—			
ANDE>	обычная точность	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.55	0.24	
		выполнена	с непрерывн.	230	172	86	34	—	—	—	92	40
			без непрерывности	234	176	88	88	—	—	—	93	
	двойная точность	не выполнена	—	—	—	—	—	—	—	—	—	
		выполнена	с непрерывн.	—	—	—	—	—	—	—	92	40
			без непрерывности	—	—	—	—	—	—	—		
ORE>	обычная точность	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.55	0.24	
		выполнена	с непрерывн.	231	174	87	34	—	—	—	93	40
			без непрерывности	234	176	88	88	—	—	—	92	
	двойная точность	не выполнена	—	—	—	—	—	—	—	—	—	
		выполнена	с непрерывн.	—	—	—	—	—	—	—	93	40
			без непрерывности	—	—	—	—	—	—	—	92	
LDE<=	обычная точность	с непрерывностью	235	177	89	34	—	—	—	93	40	
		без непрерывности	231	174	87	88	—	—	—	92		
	двойная точность	с непрерывностью	—	—	—	—	—	—	—	93	40	
		без непрерывности	—	—	—	—	—	—	—	92		
ANDE<=	обычная точность	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.55	0.24	
		выполнена	с непрерывн.	234	176	88	34	—	—	—	92	40
			без непрерывности	230	172	86	86	—	—	—		
	двойная точность	не выполнена	—	—	—	—	—	—	—	—	—	
		выполнена	без непрерывности	—	—	—	—	—	—	—	92	40
			—	—	—	—	—	—	—			
ORE<=	обычная точность	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.55	0.24	
		выполнена	с непрерывн.	234	176	88	34	—	—	—	92	40
			без непрерывности	230	172	86	86	—	—	—		
	двойная точность	не выполнена	—	—	—	—	—	—	—	—	—	
		выполнена	с непрерывн.	—	—	—	—	—	—	—	92	40
			без непрерывности	—	—	—	—	—	—	—		

Команда	Обработка (операнд)		Время обработки (мкс)									
			Серия QnA				System Q					
			Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H	
LDE<	обычная точность	с непрерывностью	231	174	87	35	—	—	—	92	40	
		без непрерывности	234	176	88	88	—	—	—			
	двойная точность	с непрерывностью	—	—	—	—	—	—	—	92	92	
		без непрерывности	—	—	—	—	—	—	—			
ANDE<	обычная точность	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.55	0.24	
		выполнена	с непрерывн.	230	172	86	34	—	—	—	92	40
			без непрерывности	234	176	88	88	—	—	—		
	двойная точность	не выполнена	—	—	—	—	—	—	—	—	—	
		выполнена	с непрерывн.	—	—	—	—	—	—	—	92	40
			без непрерывности	—	—	—	—	—	—	—		
ORE<	обычная точность	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.55	0.24	
		выполнена	с непрерывн.	231	174	87	34	—	—	—	93	40
			без непрерывности	234	176	88	88	—	—	—		
	двойная точность	не выполнена	—	—	—	—	—	—	—	—	—	
		выполнена	с непрерывн.	—	—	—	—	—	—	—	93	40
			без непрерывности	—	—	—	—	—	—	—		
LDE>=	обычная точность	с непрерывностью	235	177	89	35	—	—	—	93	40	
		без непрерывности	231	174	87	87	—	—	—	92		
	двойная точность	с непрерывностью	—	—	—	—	—	—	—	93	40	
		без непрерывности	—	—	—	—	—	—	—	92		
ANDE>=	обычная точность	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.55	0.24	
		выполнена	с непрерывн.	234	176	88	34	—	—	—	92	40
			без непрерывности	231	174	87	87	—	—	—		
	двойная точность	не выполнена	—	—	—	—	—	—	—	—	—	
		выполнена	с непрерывн.	—	—	—	—	—	—	—	92	40
			без непрерывности	—	—	—	—	—	—	—		
ORE>=	обычная точность	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.55	0.24	
		выполнена	с непрерывн.	234	176	88	34	—	—	—	92	40
			без непрерывности	230	172	86	86	—	—	—		
	двойная точность	не выполнена	—	—	—	—	—	—	—	—	—	
		выполнена	с непрерывн.	—	—	—	—	—	—	—	92	40
			без непрерывности	—	—	—	—	—	—	—		
LD\$=	с непрерывностью		97	73	37	37	—	—	—	38	16	
	без непрерывности		81	61	31	31	—	—	—	34	15	
AND\$=	не выполнена		1.4	1.1	0.55	0.55	—	—	—	0.56	0.23	
	выполнена	с непрерывностью	96	72	36	36	—	—	—	39	17	
		без непрерывности	81	61	31	31	—	—	—	32	14	
OR\$=	не выполнена		1.4	1.1	0.55	0.55	—	—	—	0.56	0.24	
	выполнена	с непрерывностью	97	73	37	37	—	—	—	40	17	
		без непрерывности	80	60	30	30	—	—	—	33	14	
LD\$<>	с непрерывностью		83	62	31	31	—	—	—	32	14	
	без непрерывности		97	73	37	37	—	—	—	40	17	
AND\$<>	не выполнена		1.4	1.1	0.55	0.55	—	—	—	0.56	0.23	
	выполнена	с непрерывностью	80	60	30	30	—	—	—	33	14	
		без непрерывности	96	72	36	36	—	—	—	39	17	
OR\$<>	не выполнена		1.4	1.1	0.55	0.55	—	—	—	0.56	0.24	
	выполнена	с непрерывностью	81	61	31	31	—	—	—	32	14	
		без непрерывности	96	72	36	36	—	—	—	39	17	

Команда	Обработка (операнд)	Время обработки (мкс)									
		Серия QnA				System Q					
		Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H	
LD\$>	с непрерывностью	83	62	31	31	—	—	—	32	14	
	без непрерывности	97	73	37	37	—	—	—	40	17	
AND\$>	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.56	0.23	
	выполнена	с непрерывностью	80	60	30	30	—	—	—	33	14
		без непрерывности	96	72	36	36	—	—	—	39	17
OR\$>	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.56	0.24	
	выполнена	с непрерывностью	81	61	31	31	—	—	—	32	14
		без непрерывности	96	72	36	36	—	—	—	39	17
LD\$<=	с непрерывностью	97	73	37	37	—	—	—	40	17	
	без непрерывности	81	61	31	31	—	—	—	32	14	
AND\$<=	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.56	0.23	
	выполнена	с непрерывностью	96	72	36	36	—	—	—	39	17
		без непрерывности	81	61	31	31	—	—	—	32	14
OR\$<=	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.56	0.24	
	выполнена	с непрерывностью	97	73	37	37	—	—	—	40	17
		без непрерывности	80	60	30	30	—	—	—	33	14
LD\$<	с непрерывностью	81	61	31	31	—	—	—	32	14	
	без непрерывности	97	73	37	37	—	—	—	40	17	
AND\$<	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.56	0.23	
	выполнена	с непрерывностью	80	60	30	30	—	—	—	32	14
		без непрерывности	96	72	36	36	—	—	—	39	16
OR\$<	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.56	0.24	
	выполнена	с непрерывностью	81	61	31	31	—	—	—	32	14
		без непрерывности	96	72	36	36	—	—	—	39	16
LD\$>=	с непрерывностью	97	73	37	37	—	—	—	40	17	
	без непрерывности	81	61	31	31	—	—	—	32	14	
AND\$>=	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.56	0.23	
	выполнена	с непрерывностью	96	72	36	36	—	—	—	39	16
		без непрерывности	81	61	31	31	—	—	—	32	14
OR\$>=	не выполнена	1.4	1.1	0.55	0.55	—	—	—	0.56	0.24	
	выполнена	с непрерывностью	97	73	37	37	—	—	—	39	17
		без непрерывности	80	60	30	30	—	—	—	32	14
ВКСМР=	n = 1	120	90	45	45	130	105	97	48	21	
ВКСМР=P	n = 96	367	276	138	138	205	175	165	142	61	
ВКСМР<>	n = 1	123	92	46	46	130	105	98	48	21	
ВКСМР<>P	n = 96	346	260	130	130	210	180	165	150	65	
ВКСМР>	n = 1	123	92	96	96	130	105	97	48	21	
ВКСМР>P	n = 96	366	275	138	138	210	180	165	142	61	
ВКСМР>=	n = 1	121	91	46	46	130	105	98	48	21	
ВКСМР>=P	n = 96	386	290	145	145	205	175	165	150	65	
ВКСМР<	n = 1	121	91	96	96	130	105	98	48	21	
ВКСМР<P	n = 96	366	275	138	138	210	180	165	158	68	
ВКСМР<=	n = 1	121	91	46	46	130	105	97	48	21	
ВКСМР<=P	n = 96	348	261	131	131	205	175	165	150	65	
+ (s, d) +P (s, d)		2.4	1.8	0.9	0.9	1.0	0.80	0.50	0.39	0.17	
+ (s1, s2, d) +P (s1, s2, d)		2.7	2.0	1.0	1.0	1.2	0.96	0.60	0.47	0.20	
− (s, d) −P (s, d)		2.4	1.8	0.9	0.9	1.0	0.80	0.50	0.39	0.17	
(s1, s2, d) −P (s1, s2, d)		2.6	2.0	1.0	1.0	1.2	0.96	0.60	0.47	0.20	

Команда	Обработка (операнд)	Время обработки (мкс)								
		Серия QnA				System Q				
		Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
D+ (s, d) D + P (s, d)		2.8	2.1	1.1	1.1	1.3	1.04	0.65	0.71	0.31
D+ (s1, s2, d) D + P (s1, s2, d)		3.2	2.4	1.2	1.2	1.5	1.2	0.75	0.79	0.34
D - (s, d) D - P (s, d)		2.8	2.1	1.1	1.1	1.3	1.04	0.65	0.71	0.30
D(s1, s2, d) D - P (s1, s2, d)		3.2	2.4	1.2	1.2	1.5	1.2	0.75	0.79	0.34
x (s1, s2, d) xP (s1, s2, d)		2.8	2.1	1.1	1.1	1.1	0.88	0.55	0.47	0.20
/ (s1, s2, d) /P (s1, s2, d)		6.8	5.1	2.6	2.6	19	16	15	2.7	1.2
Dx (s1, s2, d) D x P (s1, s2, d)		20	15	7.5	7.5	41	34	31	7.9	3.4
D/ (s1, s2, d) D/P (s1, s2, d)		36	27	13.5	13.5	28	23	21	14	6.1
B + (s, d) B + P (s, d)		5.5	4.1	2.1	2.1	34	28	26	2.2	1.0
B+ (s1, s2, d) B + P (s1, s2, d)		13	9.6	4.8	4.8	47	39	37	5.0	2.2
B - (s, d) B - P (s, d)		5.2	3.9	2.0	2.0	34	28	26	2.0	0.9
B (s1, s2, d) B - P (s1, s2, d)		13	9.4	4.7	4.7	48	40	38	4.9	2.1
DB+ (s, d) DB + P (s, d)		29	22	11	11	58	48	44	12	5.0
DB+ (s1, s2, d) DB + P (s1, s2, d)		32	24	12	12	60	49	46	12	5.3
DB (s,d) DB - P (s,d)		29	22	11	11	59	48	45	11	4.8
DB (s1, s2,d) DB - P (s1, s2, d)		32	24	12	12	60	51	45	12	5.2
Bx (s1, s2, d) B x P (s1, s2, d)		9.4	7.1	3.6	3.6	42	35	33	3.7	1.6
B/ (s1, s2, d) B/P (s1, s2, d)		9.4	7.1	3.6	3.6	48	40	37	3.8	1.6

Команда	Обработка (операнд)		Время обработки (мкс)								
			Серия QnA				System Q				
			Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
DBx (s1, s2, d) DB x P (s1, s2, d)			62	46	23	23	140	120	110	24	10
DB/ (s1, s2, d) DB/P (s1, s2, d)			69	52	26	26	83	69	65	27	12
E+ (s, d) E + P (s, d)	обычная точность	s = 0, d = 0	54	40	20	35	—	—	—	1.8	0.78
		$s = 2^{127}, d = 2^{127}$	524	394	197						
E+ (s1, s2, d) E + P (s1, s2, d)	двойная точность	s = 0, d = 0	—	—	—	—	—	—	—	203	87
		$s = 2^{127}, d = 2^{127}$	—	—	—						
E+ (s1, s2, d) E + P (s1, s2, d)	обычная точность	s1 = 0, s2 = 0	54	40	20	35	—	—	—	2.4	1.1
		$s1 = 2^{127}, s2 = 2^{127}$	524	394	197						
E+ (s1, s2, d) E + P (s1, s2, d)	двойная точность	s = 0, d = 0	—	—	—	—	—	—	—	209	90
		$s = 2^{127}, d = 2^{127}$	—	—	—						
E- (s, d) E - P (s, d)	обычная точность	s = 0, d = 0	54	40	20	35	—	—	—	1.8	0.78
		$s = 2^{127}, d = 2^{127}$	515	387	194						
E- (s1, s2, d) E - P (s1, s2, d)	двойная точность	s = 0, d = 0	—	—	—	—	—	—	—	202	87
		$s = 2^{127}, d = 2^{127}$	—	—	—						
E- (s1, s2, d) E - P (s1, s2, d)	обычная точность	s1 = 0, s2 = 0	55	41	21	36	—	—	—	2.4	1.1
		$s1 = 2^{127}, s2 = 2^{127}$	520	391	146						
E- (s1, s2, d) E - P (s1, s2, d)	двойная точность	s = 0, d = 0	—	—	—	—	—	—	—	210	90
		$s = 2^{127}, d = 2^{127}$	—	—	—						
Ex (s1, s2, d) Ex P (s1, s2, d)	обычная точность	s1 = 0, s2 = 0	55	41	21	36	—	—	—	2.4	1.1
		$s1 = 2^{127}, s2 = 2^{127}$	567	426	218						
Ex (s1, s2, d) Ex P (s1, s2, d)	двойная точность	s = 0, d = 0	—	—	—	—	—	—	—	222	96
		$s = 2^{127}, d = 2^{127}$	—	—	—						
E/ (s1, s2, d) E/P (s1, s2, d)	обычная точность	s1 = 0, s2 = 1	149	112	56	37	—	—	—	12	5.2
		$s1 = 2^{127}, s2 = -2^{126}$	1109	834	417						
E/ (s1, s2, d) E/P (s1, s2, d)	двойная точность	s = 0, d = 0	—	—	—	—	—	—	—	369	159
		$s1 = 2^{127}, s2 = -2^{126}$	—	—	—						
§+ (s, d) § + P (s, d)			179	134	67	67	—	—	—	68	29
§+ (s1, s2, d) § + P (s1, s2, d)			206	155	78	78	—	—	—	81	35
INC INCP			1.9	1.4	0.7	0.7	0.70	0.56	0.35	0.32	0.14
DINC DINCP			2.3	1.7	0.9	0.9	0.90	0.72	0.45	0.47	0.20
DEC DECP			1.9	1.4	0.7	0.7	0.70	0.56	0.35	0.32	0.14
DDEC DDECP			2.3	1.7	0.9	0.9	0.90	0.72	0.45	0.47	0.20
BCD BCDP			2.7	2.0	1.0	1.0	20	16	15	1.1	0.48
DBCD DBCDP			7.9	5.9	3.0	3.0	26	21	20	3.2	1.4
BIN BINP			2.7	2.0	1.0	1.0	19	16	15	1.0	0.44
DBIN DBINP			4.8	3.6	1.8	1.8	22	18	17	1.9	0.82

Команда	Обработка (операнд)		Время обработки (мкс)								
			Серия QnA				System Q				
			Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
INT INTP	обычная точность	s = 0	20	15	7.5	7.5	—	—	—	3.2	1.4
		s = 32766.5	54	40	20	20					
	двойная точность	s = 0	—	—	—	—	—	—	—	22	9.3
		s = 32766.5	—	—	—	—					
DINT DINTP	обычная точность	s = 0	20	15	7.5	7.5	—	—	—	2.5	1.1
		s = 1234567890.3	59	44	22	22					
	двойная точность	s = 0	—	—	—	—	—	—	—	24	10
		s = 1234567890.3	—	—	—	—					
FLT FLTP	обычная точность	s = 0	27	20	10	10	—	—	—	2.1	0.92
		s = 7FFF <sub>H</sub>	55	41	21	21					
	двойная точность	s = 0	—	—	—	—	—	—	—	22	9.6
		s = 7FFF <sub>H</sub>	—	—	—	—					
DFLT DFLTP	обычная точность	s = 0	28	21	11	11	—	—	—	2.1	0.88
		s = 7FFFFFFF <sub>H</sub>	56	42	21	21					
	двойная точность	s = 0	—	—	—	—	—	—	—	26	11
		s = 7FFFFFFF <sub>H</sub>	—	—	—	—					
DBL DBLP			12	8.6	4.3	4.3	19	16	15	4.5	1.9
WORD WORDP			12	9.0	4.5	4.5	23	19	17	4.7	2.0
GRY GRYP			12	9.0	4.5	4.5	19	16	15	4.7	2.0
DGRY DGRYP			14	10	5.0	5.0	23	19	17	5.3	2.3
GBIN GBINP			46	34	17	17	52	42	40	18	7.7
DGBIN DGBINP			83	62	31	31	110	88	84	32	14
NEG NEGP			9.3	7	3.5	3.5	16	13	12	3.6	1.6
DNEG DNEGP			11	8.2	4.1	4.1	19	17	15	4.3	1.8
ENEG ENE GP			9.8	7.4	3.7	3.7	—	—	—	3.9	1.7
BKBCD (s, d, n) BKBCDP (s, d, n)	n = 1		102	76	38	38	78	63	57	38	17
	n = 96		272	204	102	102	315	275	250	99	43
BKBIN (s, d, n) BKBINP (s, d, n)	n = 1		102	76	38	38	74	61	57	38	17
	n = 96		272	204	102	102	285	255	230	99	43
MOV MOV P	s = D0, d = D1		0.7	0.5	0.3	0.3	0.70	0.56	0.35	0.24	0.10
	s = D0, d = J1/W1		392 <sup>1</sup>	305 <sup>1</sup>	176 <sup>1</sup>	176 <sup>1</sup>	155	130	120	140	160
			391 <sup>2</sup>	299 <sup>2</sup>	165 <sup>2</sup>	165 <sup>2</sup>					
DMOV DMOV P	s = K4X0, d = D1		2.4	1.8	0.9	0.9	0.90	0.72	0.45	0.47	0.20
	s = K4X0, d = J1/W1		400 <sup>1</sup>	313 <sup>1</sup>	183 <sup>1</sup>	183 <sup>1</sup>	165	135	120	147	64
			395 <sup>2</sup>	301 <sup>2</sup>	167 <sup>2</sup>	167 <sup>2</sup>					
EMOV EMOV P			12	8.6	4.3	4.3	—	—	—	0.63	0.27
\$MOV \$MOV P			100	75	38	38	46	38	35	40	17
							98	80	73		
CMLCMLP			2.0	1.5	0.8	0.8	0.70	0.56	0.35	0.40	0.17

<sup>1</sup> Эти значения времени обработки действительны в случае применения главной монтажной шины A38B/A1S38B и расширительной монтажной шины.

<sup>2</sup> Эти значения времени обработки действительны в случае применения главной монтажной шины A38HB/A1S38HB.



Команда	Обработка (операнд)	Время обработки (мкс)								
		Серия QnA				System Q				
		Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
DCML DCMLP		2.4	1.8	0.9	0.9	0.90	0.72	0.45	0.55	0.24
BMOV (s, d, n)	n = 1	43	32	16	16	27	21	20	17	7.1
BMOV (s, d, n)	n = 96	81	61	31	31	72	62	53	32	14
FMOV (s, d, n)	n = 1	18	13	6.5	6.5	23	19	17	6.7	2.9
FMOV (s, d, n)	n = 96	36	27	14	14	48	41	36	14	6.1
XCH XCHP		3.1	2.3	1.2	1.2	7.6	6.3	5.7	1.3	0.54
DXCH DXCHP		3.1	2.3	1.2	1.2	9.5	8.0	7.1	1.3	0.54
BXCH (d1, d2, n)	n = 1	77	58	29	29	62	51	48	31	13
BXCHP (d1, d2, n)	n = 96	213	160	80	80	165	140	125	84	36
SWAP SWAPP		9.2	6.9	3.5	3.5	17	14	13	3.7	1.6
CJ		7.8	5.8	2.9	2.9	10	8.5	8.1	3.2	1.4
SCJ		7.8	5.8	2.9	2.9	10	8.5	8.1	3.2	1.4
JMP		8.0	6.0	3.0	3.0	11	8.5	8.1	3.2	1.4
GOEND		2.0	1.5	0.75	0.75	3.3	2.9	2.8	0.39	0.34
EI		3.1	2.3	1.2	1.2	14	11	11	1.3	0.54
DI		2.3	1.7	0.9	0.9	13	12	11	0.95	0.41
IMASK		8.1	6.5	3.3	3.3	41	34	35	11	4.6
IRET		4.0	3.0	1.5	1.5	205	170	155	1.6	0.68
RFS RFSF	s = X, n = 1	31.3	23.4	11.7	11.7	55	46	43	6.7	4.7
	54					45	41			
	s = X, n = 96	97.6	72.8	36.4	36.4	79	64	59	19	13
	s = Y, n = 96					73	61	56		
UDCNT1		42.6	31.8	15.9	15.9	—	—	—	15	6.5
UDCNT2		44.6	33.3	16.7	16	—	—	—	16	6.8
TTMR		25.9	19.3	9.7	9.7	—	—	—	10	4.4
STMR		41.7	31.1	15.6	15.6	—	—	—	20	7.1
ROTC		66.1	49.3	24.7	24.7	—	—	—	26	11
RAMP		45.4	33.9	17.0	17.0	—	—	—	18	7.7
SPD		48.9	36.5	18.3	18.3	—	—	—	19	8.3
PLSY		26.9	20.1	10.1	10.1	—	—	—	10	4.5
PWM		32.8	24.5	12.3	12.3	—	—	—	9.1	3.9
MTR		29.2	21.8	10.9	10.9	—	—	—	11	4.9
WAND (s, d) WANDP (s, d)		2.4	1.8	0.9	0.9	1.0	0.80	0.50	0.39	0.17
WAND (s1, s2, d) WANDP (s1, s2, d)		9.5	7.1	3.6	3.6	1.2	0.96	0.60	0.47	0.20
DAND (s, d) DANDP (s, d)		3.0	2.3	1.2	1.2	1.3	1.04	0.65	0.71	0.31
DAND (s1, s2, d) DANDP (s1, s2, d)		19	14	7.0	7.0	1.5	1.2	0.75	0.79	0.34

Команда	Обработка (операнд)	Время обработки (мкс)								
		Серия QnA				System Q				
		Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
BKAND (s1, s2, d, n) BKANDP (s1, s2, d, n)	n = 1	89	67	34	34	110	87	79	36	16
	n = 96	184	138	69	69	185	155	140	74	32
WOR (s, d) WORP (s, d)		2.4	1.8	0.9	0.9	1.0	0.80	0.50	0.40	0.17
WOR (s1, s2, d) WORP (s1, s2, d)		9.5	7.1	3.6	3.6	1.2	0.96	0.60	0.47	0.20
DOR (s, d) DORP (s, d)		3.0	2.3	1.2	1.2	1.3	1.04	0.65	0.71	0.31
DOR (s1, s2, d) DORP (s1, s2, d)		19	14	7.0	7.0	1.5	1.2	0.75	0.79	0.34
BKOR (s1, s2, d, n) BKORP (s1, s2, d, n)	n = 1	89	67	34	34	110	87	81	36	16
	n = 96	184	138	69	69	185	155	140	74	32
WXOR (s, d) WXORP (s, d)		2.4	1.8	0.9	0.9	1.0	0.80	0.50	0.39	0.17
WXOR (s1, s2, d) WXORP (s1, s2, d)		17.2	7.1	3.6	3.6	1.2	0.96	0.60	0.47	0.20
DXOR (s, d) DXORP (s, d)		3.0	2.3	1.2	1.2	1.3	1.04	0.65	0.71	0.31
DXOR (s1, s2, d) DXORP (s1, s2, d)		19	14	7.0	7.0	1.5	1.2	0.75	0.79	0.34
BKXOR (s1, s2, d, n) BKXORP (s1, s2, d, n)	n = 1	89	67	34	34	110	87	81	36	16
	n = 96	184	138	69	69	183	155	140	74	32
WXNR (s, d) WXNRP (s, d)		2.4	1.8	0.9	0.9	1.0	0.80	0.50	0.40	0.17
WXNR (s1, s2, d) WXNRP (s1, s2, d)		9.5	7.1	3.6	3.6	1.2	0.96	0.60	0.47	0.20
DXNR (s, d) DXNRP (s, d)		3.0	2.3	1.2	1.2	1.3	1.04	0.65	0.71	0.31
DXNR (s1, s2, d) DXNRP (s1, s2, d)		24	18	9	9	1.5	1.2	0.75	0.79	0.34
BKXNR (s1, s2, d, n) BKXNR (s1, s2, d, n)	n = 1	89	67	34	34	110	87	82	36	16
	n = 96	184	138	69	69	185	155	140	74	32

Команда	Обработка (операнд)	Время обработки (мкс)								
		Серия QnA				System Q				
		Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
ROR (d, n)	n = 1	5.0	3.8	1.9	1.9	13	11	9.7	2.0	0.85
RORP (d, n)	n = 15	5.0	3.8	1.9	1.9	13	11	9.7	2.0	0.85
RCR (d, n)	n = 1	4.0	3.0	1.5	1.5	15	12	12	1.6	0.68
RCRP (d, n)	n = 15	4.0	3.0	1.5	1.5	15	13	12	1.6	0.68
ROL (d, n)	n = 1	5.0	3.8	1.9	1.9	13	11	10	2.0	0.85
ROLP (d, n)	n = 15	5.0	3.8	1.9	1.9	13	11	10	2.0	0.85
RCL (d, n)	n = 1	4.0	3.0	1.5	1.5	15	13	12	1.6	0.68
RCLP (d, n)	n = 15	4.0	3.0	1.5	1.5	16	13	12	1.6	0.68
DROR (d, n)	n = 1	9.8	7.4	3.7	3.7	15	12	12	3.9	1.7
DRORP (d, n)	n = 31	10	7.8	3.9	3.9	15	13	12	4.0	1.7
DRCR (d, n)	n = 1	11	8.1	4.1	4.1	17	14	14	4.3	1.8
DRCRP (d, n)	n = 31	11	8.3	4.2	4.2	18	16	15	4.3	1.9
DROL (d, n)	n = 1	9.8	7.4	3.7	3.7	14	13	12	3.9	1.7
DROLP (d, n)	n = 31	10	7.8	3.9	3.9	14	13	12	4.0	1.7
DRCL (d, n)	n = 1	11	8.1	4.1	4.1	18	15	14	4.3	1.8
DRCLP (d, n)	n = 31	11	8.3	4.2	4.2	20	17	16	4.3	1.9
SFR (d, n)	n = 1	4.4	3.3	1.7	1.7	13	10	9.7	1.7	0.75
SFRP (d, n)	n = 15	5.0	3.8	1.9	1.9	13	11	9.5	2.0	0.85
SFL (d, n)	n = 1	4.4	3.3	1.7	1.7	12	10	9.5	1.7	0.75
SFLP (d, n)	n = 15	5.0	3.8	1.9	1.9	12	9.8	9.5	2.0	0.85
BSFLR (d, n)	n = 1	51	38	19	19	42	35	33	20	8.6
BSFLRP (d, n)	n = 96	60	45	23	23	69	58	54	24	10
BSFL (d, n)	n = 1	49	37	19	19	41	34	32	20	8.5
BSFLP (d, n)	n = 96	58	44	22	22	63	53	50	23	10
DSFR (d, n)	n = 1	3.6	2.6	1.3	1.3	19	16	15	1.3	0.58
DSFRP (d, n)	n = 96	63	47	24	24	71	61	53	25	11
DSFL (d, n)	n = 1	3.6	2.6	1.3	1.3	19	16	15	1.3	0.58
DSFLP (d, n)	n = 96	65	49	25	25	70	60	52	26	11
BSET (d, n)	n = 1	20	15	7.5	7.5	27	22	20	7.6	3.3
BSETP (d, n)	n = 15	20	15	7.5	7.5	27	22	20	7.6	3.3
BRST (d, n)	n = 1	20	15	7.5	7.5	27	22	21	7.6	3.3
BRSTP (d, n)	n = 15	20	15	7.5	7.5	27	22	21	7.6	3.3
TEST (s1, s2, d) TESTP (s1, s2, d)		21	16	8.0	8.0	35	30	27	8.2	3.5

Команда	Обработка (операнд)		Время обработки (мкс)								
			Серия QnA				System Q				
			Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
DTEST (s1, s2, d) DTESTP (s1, s2, d)			24	18	9.0	9.0	37	31	28	9.2	3.9
BKRST (s, n)	n = 1		45	34	17	17	49	41	38	18	7.8
BKRST (s, n)	n = 96		49	37	19	19	64	54	50	19	8.2
SER (s1, s2, d, n)	n = 1	соответствие	58	44	22	22	56	54	42	22	9.6
SERP (s1, s2, d, n)		соответствия нет	57	43	21	21	56	54	42	21	8.9
SER	n = 96	соответствие	293	220	110	110	280	240	220	115	49
SERP		соответствия нет	340	256	128	128	280	240	220	133	57
DSER (s1, s2, d, n)	n = 1	соответствие	61	46	23	23	71	67	53	23	9.9
DSERP (s1, s2, d, n)		соответствия нет	58	44	22	22	71	67	54	23	9.7
DSER	n = 96	соответствие	354	266	133	133	495	415	375	142	61
DSERP		соответствия нет	354	266	133	133	500	415	375	132	57
SUM	s = 0		9.8	7.4	3.7	3.7	32	26	25	3.9	1.7
SUMP	s = FFFF <sub>H</sub>						27	22	21		
DSUM	s = 0		12	9.0	4.5	4.5	54	44	42	4.7	2.0
DSUMP	s = FFFFFFFF <sub>H</sub>		31	23	12	12	54	44	42	12	5.0
DECO (s, d, n)	n = 2		48	36	18	18	60	50	46	20	8.6
DECO (s, d, n)	n = 8		62	47	24	24	80	65	61	27	12
ENCO (s, d, n)	n = 2	M1 = вкл.	52	39	20	20	66	55	51	21	9.1
ENCO (s, d, n)		M4 = вкл.	52	39	20	20	66	54	51	21	9.1
ENCOP (s, d, n)	n = 8	M1 = вкл.	65	49	25	25	90	76	71	28	12
ENCOP (s, d, n)		M256 = вкл.	65	49	25	25	76	74	71	26	11
SEG SEGP			3.2	2.4	1.2	1.2	8.0	6.8	6.1	1.3	0.54
DIS (s, d, n)	n = 1		46	34	17	17	47	39	36	18	7.7
DISP (s, d, n)	n = 4		51	38	19	19	53	43	40	19	8.3
UNI (s, d, n)	n = 1		53	40	20	20	54	44	41	21	8.9
UNIP (s, d, n)	n = 4		57	43	22	22	60	49	46	23	9.7
NDIS (s1, d, s2) NDISP (s1, d, s2)			104	78	39	39	92	76	38	41	18
NUNI (s1, d, s2) NUNIP (s1, d, s2)			105	79	40	40	47	39	36	42	18
WTOB (s, d, n)	n = 1		125	94	47	47	56	46	42	47	20
WTOBP (s, d, n)	n = 96		257	193	97	97	190	155	145	99	43
BTOW (s, d, n)	n = 1		121	91	46	46	56	46	42	45	19
BTOWP (s, d, n)	n = 96		233	175	88	88	190	155	145	89	38
MAX (s, d, n)	n = 1		43	32	16	16	48	40	36	17	7.1
MAXP (s, d, n)	n = 96		318	239	120	120	300	240	235	136	59
MIN (s, d, n)	n = 1		43	32	16	16	48	40	36	17	7.1
MINP (s, d, n)	n = 96		436	326	163	163	300	240	235	159	69

Команда	Обработка (операнд)	Время обработки (мкс)								
		Серия QnA				System Q				
		Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
DMAX (s, d, n) DMAXP (s, d, n)	n = 1	71	53	27	27	52	43	39	27	12
	n = 96	427	321	161	161	600	490	460	181	78
DMIN (s, d, n) DMINP (s, d, n)	n = 1	71	53	27	27	52	43	39	27	12
	n = 96	268	201	101	101	585	475	445	112	48
SORT (s1, n, s2, d1, d2)	n = 1	43	32	16	16	66	55	50	16	7.1
	n = 96	40*	30*	15*	15*	105	86	80	14	6.2
DSORT (s1, n, s2, d1, d2)	n = 1	44	33	17	17	98	57	52	17	7.1
	n = 96	43*	32*	16*	16*	115	96	88	16	6.8
* удлинение времени цикла в результате выполнения команды										
WSUM (s, d, n) WSUMP (s, d, n)	n = 1	41.5	31.1	15.6	15.6	52	43	40	16.4	7.1
	n = 96	173.2	129.9	65	65	175	140	135	68.4	29.5
DWSUM (s, d, n) DWSUMP (s, d, n)	n = 1	47.9	35.9	18	18	61	51	46	18.9	8.2
	n = 96	330	247.5	123.8	123.8	515	420	395	130.4	56.1
FOR	n = 0	5.2	3.9	2.0	2.0	11	8.9	8.1	2.3	1.0
NEXT		8.0	6.0	3.0	3.0	8.8	7.3	6.8	3.3	1.4
BREAK BREAKP		26	19	9.5	9.5	37	30	28	11	4.6
CALL (pn) CALLP (pn)	внутренний указатель файла	5.1	3.8	1.9	1.9	17	14	13	2.1	0.88
	общий указатель	85	64	32	32				33	14
CALL (pn s1...s5) CALLP (pn s1...s5)		348	261	131	131	245	200	190	135	58
RET	возврат в первоначальную программу	7.5	5.6	2.8	2.8	16	13	12	2.9	1.3
	возврат в другую программу	51	38	19	19	—	—	—	20	8.5
FCALL (pn) FCALLP (pn)	внутренний указатель файла	8.8	6.6	3.3	3.3	29	24	22	3.6	1.6
	общий указатель	48	36	18	18				20	8.7
FCALL (pn S1...S5) FCALLP (pn S1...S5)		388	254	127	127	250	205	190	134	57
ECALL (pn) ECALLP (pn)		187	140	70	70	—	—	—	77	33
ECALL (pn S1...S5) ECALLP (pn S1...S5)		515	387	144	144	—	—	—	162	70
EFCALL (pn) EFCALLP (pn)		188	141	71	71	—	—	—	78	34
EFCALL (pn S1...S5) EFCALLP (pn S1...S5)		516	388	194	194	—	—	—	200	86
COM		137	103	52	52	110	77	72	55	16
IX		31	23	12	12	65	54	51	12	5.2
IXEND		12	8.9	4.5	4.5	30	26	25	4.7	2.0
IXDEV	количество контактов: 1	127	95	46	46	145	120	110	48	21
	количество контактов: 14	238	179	85	85	770	630	585	93	40

Команда	Обработка (операнд)	Время обработки (мкс)									
		Серия QnA				System Q					
		Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H	
IXSET	количество контактов: 1	127	95	46	46	145	120	110	48	21	
	количество контактов: 14	238	179	85	85	770	630	585	93	40	
FIFW FIFWP	количество блоков данных: 1	27	20	10	10	36	32	28	11	4.5	
	количество блоков данных: 96										
FIFR FIFRP	количество блоков данных: 1	34	25	13	13	45	41	36	13	5.6	
	количество блоков данных: 96	79	59	30	30	93	82	70	32	14	
FPOP FPOPP	количество блоков данных: 1	46	34	17	17	40	37	32	16	7.0	
	количество блоков данных: 96										
FINS FINSR	количество блоков данных: 1	48	36	18	18	53	44	38	20	8.4	
	количество блоков данных: 96	96	72	36	36	100	89	76	36	15	
FDEL FDELP	количество блоков данных: 1	47	35	18	18	60	50	43	19	7.5	
	количество блоков данных: 96	97	73	37	37	110	95	82	39	15	
PR	SM701 ON	1 знак	83	62	31	31	—	—	—	33	11
		32 знака	123	92	46	46				48	18
	SM701 OFF	54	40	20	20	21	7.8				
PRC		400	301	151	151	—	—	—	181	16	
светодиод	при индикации	223	167	84	84	—	—	—	—	—	
	индикация завершена	79	59	30	30						
LEDC	при индикации	559	420	210	210	—	—	—	—	—	
	индикация завершена	413	310	155	155						
LEDR	без индикации → без индикации	18	13	6.5	6.5	—	—	—	0.40	0.17	
	выполнить команду LED → без индикации	205	154	77	77				103	44	
CHKST		15	11	5.5	5.5	—	—	—	5.8	2.5	
CHK (контроль на ошибки)	ошибок нет на контакте 1	61	46	23	23	—	—	—	24	10	
	ошибок нет на контакте 150	4232	3182	1591	1591				1676	721	
	ошибка на контакте 1	224	168	84	84				88	38	
CHKCIR	10 проверочных контуров	15	11	5.5	5.5	—	—	—	5.8	2.5	
SLT	все внутренние операнды	2399	1804	902	902	—	—	—	—	—	
	регистр файлов 8 кбайт	7254	5454	2727	2727						
	окончание выполнения SLT	15	11	5.5	5.5						
SLTR		1.1	0.8	0.4	0.4	—	—	—	—	—	
STRA	запуск	47	35	18	18	—	—	—	—	—	
	окончание выполнения STRA	15	11	5.5	5.5						
STRAR		1.1	0.8	0.4	0.4	—	—	—	—	—	
PTRA		15	11	5.5	5.5	—	—	—	—	—	
PTRAR		15	11	5.5	5.5	—	—	—	—	—	
PTRAXE PTRAXEP	выполнение команды	1.6	1.2	0.6	0.6	—	—	—	—	—	
	контроль программы	169	127	64	64						
BINDA BINDAP	s = 1	40	30	15	15	—	—	—	15	6.7	
	s = -32768	60	45	23	23				24	10	
DBINDA DBINDAP	s = 1	63	47	44	44	—	—	—	43	18	
	s = -2147483648	217	163	82	82				86	37	
BINHA BINHAP	s = 1	46	34	17	17	—	—	—	18	7.7	
	s = FFFF <sub>H</sub>	48	36	18	18				19	8.2	
DBINHA DBINHAP	s = 1	59	44	22	22	—	—	—	23	10	
	s = FFFFFFFF <sub>H</sub>	62	46	23	23				24	10	
BCDDA BCDDAP	s = 1	58	43	22	22	—	—	—	23	9.8	
	s = 9999	54	40	20	20				21	8.9	
DBCDDA DBCDDAP	s = 1	61	46	23	23	—	—	—	22	9.5	
	s = 99999999	75	56	28	28				29	13	
DABIN DABINP	s = 1	133	100	50	50	—	—	—	57	25	
	s = -32768	145	109	55	55				58	28	
DDABIN DDABINP	s = 1	241	181	91	91	—	—	—	92	40	
	s = -2147483648	268	201	101	101				106	46	

Команда	Обработка (операнд)	Время обработки (мкс)									
		Серия QnA				System Q					
		Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H	
HABIN	s = 1	32	24	12	12	—	—	—	13	5.8	
HABINP	s = FFFF <sub>H</sub>	38	28	14	14	—	—	—	15	6.4	
DHABIN	s = 1	54	40	20	20	—	—	—	22	9.5	
DHABINP	s = FFFFFFFF <sub>H</sub>	63	47	24	24	—	—	—	25	11	
DABCD	s = 1	36	27	14	14	—	—	—	16	6.9	
DABCDP	s = 9999	42	31	16	16	—	—	—	17	7.2	
DDABCD	s = 1	63	47	24	24	—	—	—	25	11	
DDABCDP	s = 99999999	75	56	28	28	—	—	—	29	13	
COMRD		36	27	14	14	—	—	—	40	16	
COMRDP											
LEN	1 знак	48	36	18	18	—	—	—	18	8.0	
LENP	96 знаков	229	172	86	86	—	—	—	86	37	
STR		132	99	50	50	—	—	—	53	23	
STRP											
DSTR		285	214	107	107	—	—	—	123	53	
DSTRP											
VAL		258	194	97	97	—	—	—	95	41	
VALP											
DVAL		402	302	151	151	—	—	—	166	72	
DVALP											
ESTR		1337	1005	503	503	—	—	—	564	243	
ESTRP											
EVAL	формат с плавающей запятой	242	182	91	91	—	—	—	100	43	
EVALP	экспоненциальный формат	306	230	115	115	—	—	—	127	55	
ASC (s, d, n)	n = 1	164	123	62	62	—	—	—	64	28	
ASCP (s, d, n)	n = 96	780	586	293	293	—	—	—	289	125	
HEX (s, d, n)	n = 1	161	121	61	61	—	—	—	60	26	
HEXP (s, d, n)	n = 96	826	621	311	311	—	—	—	343	148	
RIGHT (s, d, n)	n = 1	131	98	49	49	—	—	—	49	21	
RIGHTP (s, d, n)	n = 96	354	266	133	133	—	—	—	131	56	
LEFT (s, d, n)	n = 1	129	97	49	49	—	—	—	50	21	
LEFTP (s, d, n)	n = 96	354	266	133	133	—	—	—	131	56	
MIDR		141	106	53	53	—	—	—	53	23	
MIDRP											
MIDW		341	256	128	128	—	—	—	128	55	
MIDWP											
INSTR	соответствия нет	156	117	59	59	—	—	—	58	25	
INSTRP	соответствие	первое	141	106	53	53	—	—	—	55	24
		последнее	155	116	58	58	—	—	—	58	25
EMOD		1313	987	494	494	—	—	—	527	227	
EMODP											
EREXP		4423	3325	1663	1663	—	—	—	1656	713	
EREXP											
SIN	обычная точность	4921	3700	1850	35	—	—	—	115	50	
SINP	двойная точность	—	—	—	—	—	—	—	1945	837	
COS	обычная точность	6462	4858	2429	35	—	—	—	122	53	
COSP	двойная точность	—	—	—	—	—	—	—	2618	1127	
TAN	обычная точность	6515	4898	2449	38	—	—	—	123	53	
TANP	двойная точность	—	—	—	—	—	—	—	2618	1127	
ASIN	обычная точность	890	669	335	44	—	—	—	111	48	
ASINP	двойная точность	—	—	—	—	—	—	—	2491	1072	

Команда	Обработка (операнд)	Время обработки (мкс)									
		Серия QnA				System Q					
		Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H	
ACOS	обычная точность	801	602	301	44	—	—	—	115	49	
ACOSP	двойная точность	—	—	—	—	—	—	—	2367	1019	
ATAN	обычная точность	7818	5878	2939	39	—	—	—	157	68	
ATANP	двойная точность	—	—	—	—	—	—	—	3140	1352	
RAD	обычная точность	465	349	175	31	—	—	—	17	7.2	
RADP	двойная точность	—	—	—	—	—	—	—	24	10	
DEG	обычная точность	492	369	185	31	—	—	—	17	7.2	
DEGP	двойная точность	—	—	—	—	—	—	—	23	9.9	
SQR	обычная точность	4520	3398	1699	39	—	—	—	28	12	
SQRP	двойная точность	—	—	—	—	—	—	—	1812	780	
EXP EXPP	обычная точность	s = -10	5871	4414	2207	37	—	—	—	—	—
		s = 1	5950	4474	2237					129	56
	двойная точность	s = -10	—	—	—	—				—	—
		s = 1	—	—	—	—				2386	1026
LOG LOGP	обычная точность	s = 1	1191	896	448	37	—	—	—	113	49
		s = 10	6839	5142	2571					—	—
	двойная точность	s = 1	—	—	—	—				2146	924
		s = 10	—	—	—	—				—	—
RND RNDP		10	7.5	3.8	3.8	—	—	—	3.9	1.7	
SRND SRNDP		8.8	6.6	3.3	3.3	—	—	—	3.5	1.5	
BSQR BSQRP	s = 0	16	12	6.0	6.0	—	—	—	6.2	2.7	
	s = 9999	97	73	37	37				38	16	
BDSQR BDSQRP	s = 0	17	13	6.5	6.5	—	—	—	6.2	2.7	
	s = 99999999	88	66	33	33				38	16	
BSIN BSINP		30	22	11	11	—	—	—	12	5.1	
BCOS BCOSP		32	24	12	12	—	—	—	12	5.2	
BTAN BTANP		30	22	11	11	—	—	—	12	5.2	
BASIN BASINP		52	39	20	20	—	—	—	20	8.7	
BACOS BACOSP		53	40	20	20	—	—	—	21	9.0	
BATAN BATANP		56	42	21	21	—	—	—	22	9.6	
LIMIT LIMITP		24	18	9.0	9.0	34	28	26	10	4.3	
DLIMIT DLIMITP		28	21	11	11	41	34	30	11	4.7	
BAND BANDP		24	18	9.0	9.0	33	28	25	9.8	4.2	
DBAND DBANDP		28	21	11	11	40	34	30	11	4.9	
ZONE ZONEP		24	18	9.0	9.0	31	25	24	9.1	3.9	
DZONE DZONEP		28	21	11	11	37	29	28	11	4.6	
RSET RSETP		19	14	7.0	7.0	—	18	16	6.8	2.9	
QDRSET QDRSETP		322	242	121	121	—	—	—	205	88	
QCSET QCSETP		218	164	82	82	—	—	—	147	63	
DATERD DATERDP		36	27	14	14	30	25	23	13	5.5	



Команда	Обработка (операнд)	Время обработки (мкс)								
		Серия QnA				System Q				
		Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
DATEWR DATEWRP		42	31	16	16	69	57	54	15	6.4
DATE+	без прироста числа разрядов	60	45	23	23	47	39	36	13	5.4
DATE+P	с приростом числа разрядов	60	45	23	23	50	42	38	13	5.4
DATE-	без прироста числа разрядов	59	44	22	22	47	40	36	12	5.2
DATE-P	с приростом числа разрядов	60	45	23	23	50	42	38	12	5.2
SECOND SECONDP		27	20	10	10	28	24	22	10	4.5
HOUR HOURP		31	23	12	12	38	32	29	12	5.2
MSG	1 знак	7.2	5.4	2.7	2.7	—	—	—	3.0	1.3
	32 знака	7.4	5.6	2.8	2.8	—	—	—	—	—
PKEY	время считывания	51	38	19	19	—	—	—	20	8.6
	не принято	48	36	18	18	—	—	—	19	8.2
PSTOP PSTOPP		122	92	46	46	—	—	—	79	34
POFF POFFP		120	90	45	45	—	—	—	79	34
PSCAN PSCANP		122	92	46	46	—	—	—	75	32
PLOW PLOWP		124	93	47	47	—	—	—	80	34
WDT WDTP		12	8.7	4.4	4.4	18	15	14	5.9	2.6
DUTY		1.6	1.2	0.6	0.6	41	36	32	9.3	4.0
ZRRDB ZRRDBP		19	14	6.9	6.9	—	24	22	7.9	3.4
ZRWRB ZRWRBP		21	16	7.8	7.8	—	27	24	9.4	4.0
ADRSET ADRSETP		13	9.3	4.7	4.7	23	19	18	4.9	2.1
KEY		43.4	32.4	16.2	16.2	—	—	—	17	7.3
ZPUSH ZPUSHP		27.6	20.6	10.3	10.3	38	33	30	11	4.7
ZPOP ZPOPP		12.7	9.5	4.8	4.8	37	31	29	5.1	2.2
EPROMWR EPROMWRP		62.6	46.7	23.4	23.4	—	—	—	—	—
ZCOM		4296.6	3206.4	1603.2	1603.2	105	82	80	691	289
READ		770.6	575.1	287.6	287.6	—	—	—	554	260
SREAD		858.9	641.0	320.5	320.5	—	—	—	588	278
WRITE		791.9	591.0	295.5	295.5	—	—	—	582	273
SWRITE		848.6	633.3	316.6	316.6	—	—	—	625	295
SEND		575.7	429.6	214.8	214.8	—	—	—	—	—
RECV		375.9	280.5	140.3	140.3	—	—	—	—	—
REQ		527.4	393.6	196.8	196.8	—	—	—	—	—
ZNFR		982.1	732.9	366.5	366.5	—	—	—	—	—
ZNTO		989.3	738.3	369.2	369.2	—	—	—	—	—
ZNRD	MELSENET/10	598.6	446.7	223.4	223.4	—	—	—	—	—
	MELSENET (II)	649.2	484.5	242.3	242.3	—	—	—	—	—
ZNWR	MELSENET/10	614.3	458.4	229.2	229.2	—	—	—	—	—
	MELSENET (II)	665.6	496.7	248.4	248.4	—	—	—	—	—
RFRP		590.9	441.0	220.5	220.5	—	—	—	—	—
RTOP		588.8	439.4	219.7	219.7	—	—	—	—	—
S.STMODE		—	—	—	22.6	—	—	—	—	—
S.CGMODE		—	—	—	19.4	—	—	—	—	—
S.TRUCK		—	—	—	43.7	—	—	—	—	—

Команда	Обработка (операнд)		Время обработки (мкс)									
			Серия QnA				System Q					
			Q2A Q2AS	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H	
S.SPREF	головной адрес буферной памяти = 0 (C A68AD)		—	—	—	407.1	—	—	—	—	—	
	головной адрес буферной памяти = 0 (C A62DA)		—	—	—	331.4	—	—	—	—	—	
UNIRD	n = 1		—	—	—	—	96	80	74	79	34	
	n = 16		—	—	—	—	440	370	340			
TRACE	запуск		—	—	—	—	—	—	—	176	76	
	окончание выполнения STRA		—	—	—	—	—	—	—	6.3	2.7	
TRACER			—	—	—	—	—	—	—	19	8.2	
FWRITE			—	—	—	—	—	—	—	84	36	
FREAD			—	—	—	—	—	—	—	79	34	
PLOADP			—	—	—	—	—	—	—	58	25	
PUNLOADP			—	—	—	—	—	—	—	272	117	
PSWAPP			—	—	—	—	—	—	—	308	133	
RBMOV	время передачи		1 слово		—	—	—	—	—	69	29	
			1000 слов		—	—	—	—	—	580	308	
COM <sup>1</sup>	с автом. обновлением общей области памяти	размер области = 2 килослова (0.5 килослова для каждого центрального процессора)		—	—	—	—	—	—	720	660	
		размер области = 4 килослова (1 килослово для каждого центрального процессора)		—	—	—	—	—	—	860	730	
	без автом. обновления общей области памяти		—	—	—	—	—	—	—	43	20	
FROM <sup>1</sup>	считывание из общей памяти другого центрального процессора	n3 = 0		—	—	—	—	—	—	59	29	
		n3 = 1000								530	500	
	считывание из буферной памяти специального модуля <sup>2</sup>	n3 = 1	главная монтажная шина							51	24	
			расшир. монтажная шина							54	27	
		n3 = 1000	главная монтажная шина							540	480	
			расшир. монтажная шина							1100	1050	
S.TO	s2 = 1		—	—	—	—	—	—	—	74	33	
	s2 = 256		—	—	—	—	—	—	—	126	54	

<sup>1</sup> Если команда одновременно выполняется в нескольких модулях центральных процессоров многопроцессорной системы, время обработки удлинится. Эти удлинение можно рассчитать по следующим формулам.  
Если система состоит только из главной монтажной шины:  
удлинение времени выполнения [мкс] = 0.54 x (число адресов) x (количество модулей центральных процессоров)  
Если система состоит из главной монтажной шины и расширительных монтажных шин:  
удлинение времени выполнения [мкс] = 1.30 x (число адресов) x (количество модулей центральных процессоров)

<sup>2</sup> Время выполнения в модулях, которые сопоставлены центральному процессору, выполняющему команду, не отличается от времени выполнения в модулях, сопоставленных другому центральному процессору.

Команда	Обработка (операнд)	Время обработки (мкс)								
		Серия QnA				System Q				
		Q2A	Q3A	Q4A QnASH	Q4AR	Q00J	Q00	Q01	Q2	Qn(P)H
FROM (n1, n2, d, n3) FROMP (n1, n2, d, n3)	n3 = 1	253 <sup>1</sup>	217 <sup>1</sup>	160 <sup>1</sup>	160 <sup>1</sup>	—	—	—	—	—
		252 <sup>2</sup>	210 <sup>2</sup>	154 <sup>2</sup>	154 <sup>2</sup>	—	—	—	—	—
		—	—	—	—	125 <sup>3</sup>	105 <sup>3</sup>	93 <sup>3</sup>	47 <sup>4</sup>	22 <sup>4</sup>
	n3 = 1000	4514 <sup>1</sup>	4286 <sup>1</sup>	4150 <sup>1</sup>	4150 <sup>1</sup>	—	—	—	—	—
		2855 <sup>2</sup>	2127 <sup>2</sup>	2038 <sup>2</sup>	2038 <sup>2</sup>	—	—	—	—	—
		—	—	—	—	740 <sup>3</sup>	695 <sup>3</sup>	685 <sup>3</sup>	476 <sup>4</sup>	437 <sup>4</sup>
DFRO (n1, n2, d, n3) DFROP (n1, n2, d, n3)	n3 = 1	260 <sup>1</sup>	221 <sup>1</sup>	165 <sup>1</sup>	165 <sup>1</sup>	—	—	—	—	—
		257 <sup>2</sup>	214 <sup>2</sup>	156 <sup>2</sup>	156 <sup>2</sup>	—	—	—	—	—
		—	—	—	—	130 <sup>3</sup>	110 <sup>3</sup>	100 <sup>3</sup>	51 <sup>4</sup>	24 <sup>4</sup>
	n3 = 500	4543 <sup>1</sup>	4271 <sup>1</sup>	4082 <sup>1</sup>	4082 <sup>1</sup>	—	—	—	—	—
		2883 <sup>2</sup>	2129 <sup>2</sup>	2064 <sup>2</sup>	2064 <sup>2</sup>	—	—	—	—	—
		—	—	—	—	745 <sup>3</sup>	695 <sup>3</sup>	675 <sup>3</sup>	478 <sup>4</sup>	437 <sup>4</sup>
TO (n1, n2, d, n3) TOP (n1, n2, d, n3)	n3 = 1	276 <sup>1</sup>	217 <sup>1</sup>	162 <sup>1</sup>	162 <sup>1</sup>	—	—	—	—	—
		254 <sup>2</sup>	211 <sup>2</sup>	154 <sup>2</sup>	154 <sup>2</sup>	—	—	—	—	—
		—	—	—	—	120 <sup>3</sup>	105 <sup>3</sup>	92 <sup>3</sup>	48 <sup>4</sup>	20 <sup>4</sup>
	n3 = 1000	4500 <sup>1</sup>	4319 <sup>1</sup>	4188 <sup>1</sup>	4188 <sup>1</sup>	—	—	—	—	—
		2878 <sup>2</sup>	2155 <sup>2</sup>	2043 <sup>2</sup>	2043 <sup>2</sup>	—	—	—	—	—
		—	—	—	—	735 <sup>3</sup>	680 <sup>3</sup>	645 <sup>3</sup>	479 <sup>4</sup>	412 <sup>4</sup>
DTO (n1, n2, d, n3) DTOP (n1, n2, d, n3)	n3 = 1	260 <sup>1</sup>	221 <sup>1</sup>	165 <sup>1</sup>	165 <sup>1</sup>	—	—	—	—	—
		257 <sup>2</sup>	216 <sup>2</sup>	157 <sup>2</sup>	157 <sup>2</sup>	—	—	—	—	—
		—	—	—	—	130 <sup>3</sup>	110 <sup>3</sup>	99 <sup>3</sup>	50 <sup>4</sup>	23 <sup>4</sup>
	n3 = 500	4471 <sup>1</sup>	4315 <sup>1</sup>	4198 <sup>1</sup>	4198 <sup>1</sup>	—	—	—	—	—
		2819 <sup>2</sup>	2172 <sup>2</sup>	2062 <sup>2</sup>	2062 <sup>2</sup>	—	—	—	—	—
		—	—	—	—	740 <sup>3</sup>	680 <sup>3</sup>	640 <sup>3</sup>	457 <sup>4</sup>	416 <sup>4</sup>

<sup>1</sup> Эти значения времени обработки действительны в случае применения главной монтажной шины A38B/A1S38B и расширительной монтажной шины.

<sup>2</sup> Эти значения времени обработки действительны в случае применения главной монтажной шины A38HB/A1S38HB.

<sup>3</sup> Время обработки зависит от типа расширительной монтажной шины, количества слотов и количества установленных модулей.

<sup>4</sup> Эти значения времени обработки действительны в случае применения главной монтажной шины Q312B и модуля QJ71C24 в слоте 0.

### А.2.2 Время обработки в контроллерах MELSEC серии "А"

Время обработки команды зависит от применяемого метода обработки входных и выходных сигналов:

- непосредственная обработка = ○
- обработка после отображения процесса = ○○

Команда	Обработка (операнд)		Время обработки (мкс)							
			A	AnN, AnS		A2A, A2AS	A3A			
			○	○○	○	○○	○○			
			без X, Y	с X, Y						
LD	X		2.3	1.0	2.3	0.2	0.15			
LDI										
AND										
ANI	Y, M, L, B, F, T, C		1.3	1.0	1.0	0.2	0.15			
OR										
ORI										
ANB										
ORB										
MPS			1.3	1.0	1.0	0.2	0.15			
MRD										
MPP										
OUT	Y	неизменное состояние	2.3	1.0	2.3	0.4	0.3			
		меняющееся состояние (выкл./вкл.)	2.3	1.0	2.3	0.4	0.3			
	M (кроме спец. марк.)	неизменное состояние	1.3	1.0	1.0	0.4	0.3			
		меняющееся состояние (выкл./вкл.)	1.3	1.0	1.0	0.4	0.3			
	специальные маркеры			37	37	0.8	0.6			
	F	не выполнена		66	AnN: 61 AnS: 62	61	6.6	5.0		
		выполнена	индикация	700	AnN: 633 AnS: 270	AnN: 633 AnS: 267	99	74		
	индикация завершена									
	T	время выполнения команды		1.3	1.0	1.0	0.4	0.3		
		обработка команды END	не выполнена		1.3	0	0	0.23	0.18	
			выполнена	после истечения		15	11	11	4.5	3.3
				дополнительно с	K	30	24	24	7.7	5.7
			D		36	30	30	8.3	6.2	
		C	время выполнения команды		1.3	1.0	1.0	0.4	0.3	
	обработка команды END		не выполнена		1.3	0	0	0.27	0.2	
			выполнена	без счета		14	0	0	0.27	0.2
				после истечения		14	0	0	0.27	0.2
			дополнительно с	K	28	25	25	4.2	3.1	
	D			33	30	30	4.8	3.6		

Команда	Обработка (операнд)		Время обработки (мкс)						
			A	AnN, AnS		AZA, AZAS	AZA		
				без X, Y	с X, Y				
SET	Y	не выполнена	2.3	1.0	2.3		0.4	0.3	
		выполнена	неизменное состояние	2.3	1.0	2.3		0.4	0.3
			меняющееся состояние	2.3	1.0	2.3		0.4	0.3
	M, L, S, B	выполнена	не выполнена	3.7	1.0	1.0		0.4	0.3
			неизменное состояние	41	1.0	1.0		0.4	0.3
			меняющееся состояние	41	1.0	1.0		0.4	0.3
	спец. маркеры M, B	не выполнена			2.0	3.0		0.8	0.6
		выполнена			32	32		0.8	0.6
	F	не выполнена		3.7	AnN: 3.0 AnS: 2.7	AnN: 3.0 AnS: 3.2		2.0	1.5
		выполнена	индикация	730	AnN: 638 AnS: 232	AnN: 638 AnS: 237		99	74
индикация завершена									
RST	Y	не выполнена	2.3	1.0	2.3		0.4	0.3	
		выполнена	неизменное состояние	2.3	1.0	2.3		0.4	0.3
			меняющееся состояние (выкл./вкл.)	2.3	1.0	2.3		0.4	0.3
	M, L, S, B	выполнена	не выполнена	3.7	1.0	1.0		0.4	0.3
			неизменное состояние	41	1.0	1.0		0.4	0.3
			меняющееся состояние (выкл./вкл.)	41	1.0	1.0		0.4	0.3
	спец. маркеры M, B	не выполнена			3.0	3.0		0.8	0.6
		выполнена			32	32		0.8	0.6
	F	не выполнена		3.7	AnN: 3.0 AnS: 3.6	3.0		2.0	1.5
		выполнена	индикация	680	AnN: 477 AnS: 296	AnN: 477 AnS: 283		150	112
			индикация завершена						
	T, C	не выполнена		3.7	3.0	3.0		1.4	1.1
		выполнена		57	43	43		5.6	4.2
	D, W, AO, A1, V, Z	не выполнена		3.7	3.0	3.0		1.4	1.1
		выполнена		34	28	28		8.4	6.3
	R	не выполнена		3.7	3.0	3.0		1.4	1.1
		выполнена		41	35	35		4.6	3.5
	PLS PLF	Y	не выполнена	65	59	61		2.2	1.7
выполнена			вкл.	68	62	63		2.2	1.7
			выкл.	64	60	62		2.2	1.7
M, L, B, F		не выполнена		64	59	59		2.2	1.7
		выполнена	вкл.	67	62	62		2.2	1.7
			выкл.	63	61	61		2.2	1.7
SFT SFTP	Y	не выполнена	3.7	3.0	3.0		1.4	1.1	
		выполнена	49	38	39		4.4	3.3	
	M, L, B, F	не выполнена	3.7	3.0	3.0		1.4	1.1	
		выполнена	48	38	38		4.4	3.3	
MC	Y	не выполнена	85	43	44		1.2	0.9	
		выполнена	50	39	41		1.2	0.9	
	M, L, S, B, F	не выполнена	84	43	43		1.2	0.9	
		выполнена	49	39	39		1.2	0.9	
MCR			35	26	26		0.6	0.45	
FEND END	M9084 = выкл.		2400	2150	2150		435	327	
	M9084 = вкл.		2400	2060	2060		285	214	
NOP			1.3	1.0	1.0		0.2	0.15	
LD=			95	70	70	87	3.8	2.9	
AND=			96	61	62	81	2.6	2.0	

Команда	Обработка (операнд)	Время обработки (мкс)					
		A	AnN, AnS		AZA, AZAS	AZA	
			○	○○			○
				без X, Y	с X, Y		
OR=		94	67	66	85	2.8	2.1
LD<>		98	69	69	86	4.1	3.1
AND<>		92	60	60	79	2.6	2.0
OR<>		96	66	66	84	2.8	2.1
LD>		96	67	67	84	4.1	3.1
AND>		92	60	60	79	2.6	2.0
OR>		98	66	65	83	2.8	2.1
LD<=		100	71	71	88	4.1	3.1
AND<=		94	61	61	80	2.6	2.0
OR<=		100	69	68	86	2.8	2.1
LD<		96	69	69	86	4.1	3.1
AND<		92	59	60	79	2.6	2.0
OR<		96	66	65	84	2.8	2.1
LD>=		100	71	71	88	4.1	3.1
AND>=		94	61	61	81	2.6	2.0
OR>=		100	69	68	86	2.8	2.1
LDD=		238	133	134	119	10	7.7
ANDD=		231	124	125	210	5.9	4.4
ORD=		236	133	133	218	6.3	4.7
LDD<>		235	131	132	217	10	7.7
ANDD<>		239	129	129	215	5.9	4.4
ORD<>		234	129	129	214	6.1	4.6
LDD>		238	133	133	219	9.7	7.3
ANDD>		240	131	131	217	5.8	4.4
ORD>		236	131	130	219	6.0	4.5
LDD<=		244	137	136	222	9.7	7.3
ANDD<=		238	127	128	213	5.8	4.4
ORD<=		246	137	136	221	6.0	4.5
LDD<		238	133	133	219	9.7	7.3
ANDD<		241	131	131	217	5.8	4.4
ORD<		236	131	130	215	6.0	4.5
LDD>=		243	137	137	222	9.7	7.3
ANDD>=		238	127	128	213	5.8	4.4
ORD>=		246	137	136	221	6.0	4.5
+ (s, d)		72	44	45	59	2.8	2.1
+P (s,d)							
+ (s1, s2, d) +P (s1, s2, d)		112	77	77	103	3.2	2.4
− (s, d)		74	45	45	59	2.8	2.1
−P (s, d)							
(s1, s2, d) −P (s1, s2, d)		123	79	79	107	3.2	2.4
D+ (s, d) D + P (s,d)		110	69	69	90	4.0	3.0
D+ (s1, s2, d) D + P (s1, s2, d)		140	99	99	246	4.6	3.5
D − (s, d) D − P(s, d)		110	69	69	90	4.0	3.0

Команда	Обработка (операнд)	Время обработки (мкс)					
		A	AnN, AnS		AZA, AZAS	AZA	
		○	○○	○	○○	○○	
		безX, Y	сX, Y				
D (s1, s2, d) D – P (s1, s2, d)		141	99	99	130	4.6	3.5
x (s1, s2, d) x P (s1, s2, d)		135	94	95	168	3.4	2.6
/ (s1, s2, d) /P (s1, s2, d)		144	102	103	99	11	8.6
D x (s1, s2, d) D x P (s1, s2, d)		429	341	340	370	20	15
D/ (s1, s2, d) D/P (s1, s2, d)		289	393	394	412	36	27
B+ (s, d) B + P (s, d)		210	123	123	183	6.4	4.8
B + (s1, s2, d) B + P (s1, s2, d)		217	129	129	192	6.2	4.7
B– (s, d) B – P (s, d)		210	125	125	185	34	25
B (s1, s2, d) B – P (s1, s2, d)		212	133	133	203	32	23
DB + (s, d) DB + P (s, d)		320	175	176	280	14	11
DB+ (s1, s2, d) DB + P (s1, s2, d)		321	187	186	294	14	11
DB (s, d) DB – P (s, d)		318	175	175	280	31	23
DB (s1, s2, d) DB – P (s1, s2, d)		322	185	186	294	29	22
B x (s1, s2, d) B x P (s1, s2, d)		410	299	300	358	14	11
B/ (s1, s2, d) B/P (s1, s2, d)		422	235	236	274	89	67
DB x (s1, s2, d) DB x P (s1, s2, d)		1158	941	939	1044	11	8.0
DB/ (s1, s2, d) DB/P (s1, s2, d)		998	896	894	954	62	47

<sup>1</sup> обычная точность

<sup>2</sup> двойная точность

Команда	Обработка (операнд)	Время обработки (мкс)					
		A	AnN, AnS		AZA, AZAS	AZA	
			без X, Y	с X, Y			
INC INCP		46	29	29	38	2.0	1.5
DINC DINCP		66	42	42	132	2.4	1.8
DEC DECP		48	31	31	39	2.0	1.5
DDEC DDECP		66	42	42	54	2.4	1.8
BCD BCDP		110	82	83	90	3.0	2.3
DBCD DBC DP		329	219	220	284	13	9.5
		329	219	220	284	13	9.5
BIN BINP		104	79	78	86	3.0	2.3
DBIN DBINP		311	215	216	280	6.0	4.5
NEG NEGP		105	50	49	86	8.6	6.5
MOV MOV P		72	47	47	57	17	13
DMOV DMOV P		104	67	67	87	20	15
CML CMLP		68	43	43	57	2.4	1.8
DCML DCMLP		130	74	75	108	3.2	2.4
BMOV (s1, s2, n) BMOV P (s1, s2, n)		7498	399	400	7144	1444	1083
FMOV (s1, s2, n) FMOV P (s1, s2, n)		1118	229	228	1029	1427	1070
XCH XCHP		102	60	61	84	2.8	2.1
DXCH DXCHP		170	107	107	141	4.2	3.2
CJ	без индексации	49	39	39		6.6	5.0
	с индексацией		48	48		6.6	5.0
SCJ	без индексации	54	71	71		6.6	5.0
	с индексацией		81	81		6.6	5.0
JMP		50	39	39		6.6	5.0
EI		195	38	38		3.0	2.3
DI		46	66	66		3.2	2.4
IRET		249	120	120		3.4	2.6
WAND (s, d) WANDP (s, d)		90	60	59	72	2.8	2.1
WAND (s1, s2, d) WANDP (s1, s2, d)		179	96	96	152	7.6	5.7



Команда	Обработка (операнд)	Время обработки (мкс)					
		A	AnN, AnS		AZA, AZAS	AZA	
		○	○○	○	○○	○○	
			без X, Y	с X, Y			
DAND (s, d) DANDP (s, d)		276	140	139	240	13	9.5
DAND (s1, s2, d) DANDP (s1, s2, d)							
WOR (s, d) WORP (s, d)		90	61	60	72	2.8	2.1
WOR (s1, s2, d) WORP (s1, s2, d)		176	97	96	152	7.6	5.7
DOR (s, d) DORP (s, d)		276	140	139	240	13	9.5
DOR (s1, s2, d) DORP (s1, s2, d)							
WXOR (s, d) WXORP (s, d)		91	60	59	72	2.8	2.1
WXOR (s1, s2, d) WXORP (s1, s2, d)		178	97	96	152	7.6	5.7
DXOR (s, d) DXORP (s, d)		274	140	139	240	13	9.5
DXOR (s1, s2, d) DXORP (s1, s2, d)							
WXNR (s, d) WXNRP (s, d)		89	64	62	74	3.0	2.3
WXNR (s1, s2, d) WXNRP (s1, s2, d)		177	98	96	152	7.8	5.9
DXNR (s, d) DXNRP (s, d)		277	142	140	241	15	11
DXNR (s1, s2, d) DXNRP (s1, s2, d)							
ROR (n) RORP (n)	n = 3/5 (n = 5 для всех процессоров AnN, n = 3 для всех иных процессоров кроме AnN)	66	52	51	51	5.8	4.4
RCR (n) RCRP (n)	n = 3/5 (n = 5 для всех процессоров AnN, n = 3 для всех иных процессоров кроме AnN)	74	59	59	59	6.4	4.8

Команда	Обработка (операнд)	Время обработки (мкс)					
		A	AnN, AnS		AZA, AZAS	AZA	
		○	○○	○	○○	○○	
		без X, Y	с X, Y				
ROL (n) ROLP (n)	n = 3/5 (n = 5 для всех процессоров AnN, n = 3 для всех иных процессоров кроме AnN)	68	54	53	53	5.8	4.4
RCL (n) RCLP (n)	n = 3/5 (n = 5 для всех процессоров AnN, n = 3 для всех иных процессоров кроме AnN)	74	57	57	57	6.4	4.8
DROR (n) DRORP (n)	n = 3/5 (n = 5 для всех процессоров AnN, n = 3 для всех иных процессоров кроме AnN)	97	70	70	69	11	8.3
DRCR (n) DRCRP (n)	n = 3/5 (n = 5 для всех процессоров AnN, n = 3 для всех иных процессоров кроме AnN)	95	72	72	72	12	9.2
DROL (n) DROLP (n)	n = 3/5 (n = 5 для всех процессоров AnN, n = 3 для всех иных процессоров кроме AnN)	101	70	69	69	10	7.8
DRCL (n) DRCLP (n)	n = 3/5 (n = 5 для всех процессоров AnN, n = 3 для всех иных процессоров кроме AnN)	98	68	68	68	12	8.7
SFR (s, n) SFRP (s, n)	n = 3/5 (n = 5 для всех процессоров AnN, n = 3 для всех иных процессоров кроме AnN)	102	74	72	83	5.0	3.8
SFL (d, n) SFLP (d, n)	n = 5	106	74	73	84	4.8	3.6
BSFR (d, n) BSFRP (d, n)	n = 5	145	124	123	124	29	22
BSFL (d, n) BSFLP (d, n)	n = 5	158	134	133	134	28	21
DSFR (d, n) DSFRP (d, n)	n = 5	133	118	116	—	18.8	14.1
DSFL (d, n) DSFLP (d, n)	n = 5	134	118	17	—	22	17
BSET (d, n) BSETP (d, n)	n = 5	107	90	90	—	9.6	7.2
BRST (d, n) BRSTP (d, n)	n = 5	114	97	96	—	9.6	7.2
SER (s1, s2, d, n) SERP (s1, s2, d, n)	n = 5	230	200	200	—	33	25
SUM SUMP		164	115	114	131	15	11
DSUM DSUMP		267	200	199	231	34	25

Команда	Обработка (операнд)	Время обработки (мкс)					
		A	AnN, AnS		AZA, AZAS	AZA	
		○	○○	○	○○	○○	
			без X, Y	с X, Y			
DECO (s, d, n) DECOP (s, d, n)	n = 2	249	164	163	216	32	24
ENCO (s, d, n) ENCOP (s, d, n)	n = 2	478	164	163	195	41	31
SEG		170	91	91	155	6.4	4.8
DIS (s, d, n)	n = 4	180	154	153	—	25	19
DISP (s, d, n)	n = 4	180	154	153	120	25	19
UNI (s, d, n)	n = 4	159	131	131	—	31	24
UNIP (s, d, n)	n = 4	159	131	131	—	31	24
FOR		64	53	53		5.8	4.4
NEXT		2532	41	41		8.0	6.0
CALL (pn)	без индексации	74	74	74		10	7.8
	с индексацией		78	78		10	7.8
CALLP (pn)	без индексации	74	70	70		10	7.8
	с индексацией		78	78		10	7.8
RET		249	50	50		7.0	5.3
CHG	M9084 = выкл.	8546	2420	2420		—	—
	M9084 = вкл.	—	2340	2340		—	—
SUB SUBP	без индексации	90	79	79		—	—
	с индексацией	—	85	85		—	—
FIFW FIFWP		340	101	101	123	20	15
FIFR FIFRP		202	118	118	134	69	52
PR		—	226	226	226	74	56
PRC		—	141	141	141	37	28
LED		170	203	203	203	100	75
LEDC		210	265	265	265	142	106
LEDR		520	638	638	638	106	80
LEDA		170	202	202	202	—	—
LEDB		172	211	211	211	—	—
CHK (контроль на ошибки)	1 входной контакт	—	—	AnN: 771 AnS: 240		33	25
	50 входных контактов	—	—	AnN: 3380 AnS: 3905		1257	943
	100 входных контактов	—	—	AnN: 6687 AnS: 7820		2503	1877
	150 входных контактов	—	—	AnA: 10137 AnS: 11470		3753	2815
CHK (выработка триггера)		—	121	121	121	—	—
SLT	память операндов	—	8448	8448	8448	2915	2186
	память операндов + R	—	24598	24598	24598	9996	7497
SLTR		—	29	29	29	6.6	5.0
STRA		—	30	30	30	5.0	3.8
STRAR		—	28	28	28	5.0	3.8
STC		—	28	28	28	2.4	1.8
CLC		—	31	31	31	2.4	1.8

Команда	Обработка (операнд)	Время обработки (мкс)					
		A	AnN, AnS		A2A, A2AS	A3A	
		○	○	○		○	○
		без X, Y	с X, Y				
ASC		140	120	120	120	3.4	2.6
WDT WDTP		—	64	64	64	5.0	3.8
DUTY		—	68	68	68	14	11
LRDP (n1, s, d, n2)	n2 = 1	—	190	190	190	42	32
	n2 = 32		190	190	190	42	32
LWTP (n1, d, s, n2)	n2 = 1	—	200	200	200	49	37
	n2 = 32		446	446	446	89	66
RFRP (n1, n2, d, n3)	n3 = 1	—	172	172	172	32	24
	n3 = 32		172	172	172	63	47
RTOP (n1, n2, s, n3)	n3 = 1	—	176	176	176	68	51
	n3 = 32		176	176	176	34	26

Команда	Обработка (операнды серии "А")	Время обработки (мкс)						
		A	AnN, AnS		A2A, A2AS, A2U		A3A, A3U	
		○	○ + ○		○		○	
		без X, Y	с X, Y	без X, Y	с X, Y	без X, Y	с X, Y	
FROM (n1, n2, d, n3)	n3 = 1	—	439	524	237	261	178	196
FROMP (n1, n2, d, n3)	n3 = 1000/112 <sup>1</sup>	—	6609	2358	5749	2789	4312	2092
DFRO (n1, n2, d, n3)	n3 = 1	—	449	529	244	266	183	199
DFROP (n1, n2, d, n3)	n3 = 500/56 <sup>2</sup>	—	6609	2109	5669	1669	4252	1252
TO (n1, n2, d, n3)	n3 = 1	—	449	539	243	266	182	200
TOP (n1, n2, d, n3)	n3 = 1000/112 <sup>1</sup>	—	6609	3918	5773	2117	4330	1588
DTO (n1, n2, d, n3)	n3 = 1	—	454	544	240	266	180	199
DTOP (n1, n2, d, n3)	n3 = 500/56 <sup>2</sup>	—	6609	1609	5747	1501	4310	1126

<sup>1</sup> Центральные процессоры без X и Y: n3 = 1000; прочие центральные процессоры с X и Y: n3 = 112

<sup>2</sup> Центральные процессоры без X и Y: n3 = 500; прочие центральные процессоры с X и Y: n3 = 56.

## А.3 Сравнение центральных процессоров

В этом разделе в табличном виде перечислены возможности отдельных центральных процессоров (Q, Q4AR, QnA, AnU, AnA, AnN, AnS), т. е. применимые операнды, режимы обработки, специальные маркеры и т. п.

### А.3.1 Применимые операнды

Операнды		System Q		Серия Q		Серия А			
		Q00J, Q00, Q01	Qn, QnH, QnPH	Q4AR	QnA	AnU	AnA	AnN	AnS
Количество входов и выходов		Q00J: 256 Q00: 1024 Q01: 1024	4096 для Q02(H), Q06H, Q12H, Q12PH, Q25H и Q25PH	4096	Q2A: 512 Q2A-S1: 1024 Q3A: 2048 Q4A: 4096	A2U: 512 A2U-S1: 1024 A3U: 2048 A4U: 4096	A2A: 512 A2A-S1: 1024 A3A: 2048—	A1N: 256 A2N: 512 A2N-S: 1024 A3N: 2048—	A1S: 256 A1S-S1: 512 A2S: 512 A2S-S1: 1024
Маркеры		8192 <sup>1</sup>		8192	8192 <sup>1</sup>	всего 8192	всего 8192	всего 2048	256
Фиксируемые маркеры		2048 <sup>1</sup>	8192 <sup>1</sup>	8192	8192 <sup>1</sup>				1048 <sup>1</sup>
Шаговые маркеры	основная программа	—		8192	—				0 <sup>1</sup>
	SFC	2048	8192		8192	—	—	—	
Маркеры ошибки		1024 <sup>1</sup>	2048 <sup>1</sup>	2048	2048 <sup>1</sup>	2048	2048	256	256
Маркеры, управляемые по фронту		1024 <sup>1</sup>	2048 <sup>1</sup>	2048	2048 <sup>1</sup>	—	—	—	—
Маркеры связи		2048 <sup>1</sup>	8192 <sup>1</sup>	8192	8192 <sup>1</sup>	8192	4096	1024	1024
Специальные маркеры связи		1024	2048 <sup>1</sup>	2048	2048 <sup>1</sup>	56	56	56	—
Таймеры		512 <sup>1</sup>	2048 <sup>1</sup>	2048	2048 <sup>1</sup>	всего 2048	всего 2048	всего 256	256 <sup>1</sup>
Фиксируемые таймеры		0 <sup>1</sup>	0 <sup>1</sup>	0 <sup>1</sup>	0 <sup>1</sup>				0 <sup>1</sup>
Счетчики		512 <sup>1</sup>	1024 <sup>1</sup>	1024	1024 <sup>1</sup>	1024	1024	256	256 <sup>1</sup>
Регистры данных		11136 <sup>1</sup>	12288 <sup>1</sup>	12288	12288 <sup>1</sup>	8192	6144	1024	1024
Регистры связи		2048 <sup>1</sup>	8192 <sup>1</sup>	8192	8192 <sup>1</sup>	8192	4096	1024	1024
Специальные регистры связи		1024 <sup>1</sup>	2048 <sup>1</sup>	2048	2048 <sup>1</sup>	56	56	56	—
Функциональный вход		16 (FX0 ... FXF)	16 (FX0 ... FXF)	16 (FX0 ... FXF)	16 (FX0 ... FXF)	—	—	—	—
Функциональный выход		16 (FY0 ... FYF)	16 (FY0 ... FYF)	16 (FY0 ... FYF)	16 (FY0 ... FYF)	—	—	—	—
Маркеры диагностики		1024	2048	2048	2048	256	256	256	256
Функциональные регистры		5 (FD0 ... FD4)	16 (FD0 ... FD15)	16 (FD0 ... FD15)	16 (FD0 ... FD15)	—	—	—	—

Операнды		System Q		Серия Q		Серия A			
		Q00J, Q00, Q01	Qn, QnH, QnPH	Q4AR	QnA	AnU	AnA	AnN	AnS
Регистры диагностики		1024	2048	2048	2048	256	256	256	256
Непосредственно адресуемые операнды связи		обозначенные J□□□		обозначенные J□□□		—	—	—	—
Непосредственно адресуемые операнды		обозначенные U□□\G□□		обозначенные U□□□\G□□□		—	—	—	—
Индексные регистры	Z	10 (Z0 ... Z15)	16 (Z0 ... Z15)	16 (Z0 ... Z15)	16 (Z0 ... Z15)	7 (Z, Z1 ... Z6)	7 (Z, Z1 ... Z6)	1 (Z)	1 (Z)
	V <sup>2</sup>	—		—		7 (V, V1 ... V6)	7 (V, V1 ... V6)	1 (V)	1 (V)
Регистры файлов		Q00J CPU: 0 Процессоры Q00 и Q01: 32767	32767 на каждый блок (R0 ... R32767)  1042432 (ZRO ... ZR1042432)	32767 на каждый блок (R0 ... R32767)  1042432 (ZRO ... ZR1042432)	32767 на каждый блок (R0 ... R32767)  1042432 (ZRO ... ZR1042432)	8192 на каждый блок (R0 ... R8191)	8192 на каждый блок (R0 ... R8191)	8192 на каждый блок (R0 ... R8191)	0 <sup>1</sup>
Сумматоры <sup>3</sup>		—		—		2	2	2	2
Вложения		15	15	15	15	8	8	8	8
Указатели		300	4096	4096	4096	256	256	256	256
Указатели прерывания		128	256	48	48	32	32	32	32
Блоки на языке SFC		—	320	320	320	—	—	—	—
Операнды перехода CFS		—	512	512	512	—	—	—	—
Десятичные константы		от K-2147483648 до K2147483647		от K-2147483648 до K2147483647		от K-2147483648 до K2147483647			
Шестнадцатеричные константы		от H0 до HFFFFFFF		от H0 до HFFFFFFF		от H0 до HFFFFFFF			
Константы в виде чисел с плавающей запятой		—	от E±1,17549-38 до E±3,40282+38	от E±1,17549-38 до E±3,40282+38		—	—	—	—
Строковые величины		„QnA CPU“, „ABCD“ <sup>4</sup>	„QnA CPU“, „ABCD“	„QnA CPU“, „ABCD“		—	—	—	—

<sup>1</sup> Количество адресов операндов можно изменить в параметрах.

<sup>2</sup> Центральные процессоры серий Q и QnA используют V в качестве маркера фронта.

<sup>3</sup> Команды, использующие сумматоры в центральных процессорах AnN, AnAs и AnU, имеют иной формат, чем в центральных процессорах серий Q и QnA.

<sup>4</sup> В центральных процессорах Q00J, Q00 и Q01 использование возможно только в сочетании с командой \$MOV.

### А.3.2 Режимы обработки ввода-вывода

Режим обработки ввода-вывода		Тип центрального процессора					
		Q	QnA/ QnAR	AnU	AnA	AnN	
Обработка	после отображения	●		●	●	● <sup>2</sup>	
	Непосредственная обработка входов и выходов	Команда частичного обновления	●		●	●	●
		Расширенные команды <sup>1</sup>	—		●	●	—
		Непосредств. адресуемые входы	●		—	—	—
		Непосредств. адресуемые выходы	●		—	—	—
Непосредственная обработка		—		—	—	● <sup>2</sup>	

<sup>1</sup> Команды DOUT, DSET и SRST являются специализированными командами для непосредственно адресуемых выходов. Для непосредственно адресуемых входов никаких специализированных команд не имеется.

<sup>2</sup> В модуле центрального процессора AnN переключение между обработкой после отображения и непосредственной обработкой осуществляется с помощью микровыключателя.

### А.3.3 Типы данных

Установленные данные		Q	QnA / QnAR	AnU	AnA	AnN
Битовые данные	битовый операнд	●		●	●	●
	словный операнд	● (необходима адресация битов)		—	—	—
16-битное слово данных	битовый операнд	● (указать количество разрядов)		● (указать количество разрядов)	● (количество разрядов установить)	● (указать количество разрядов)
	словный операнд	●		●	●	●
32-битное слово данных	битовый операнд	● (указать количество разрядов)		● (указать количество разрядов)	● (указать количество разрядов)	● (указать количество разрядов)
	словный операнд	●		●	●	●
Числа с плавающей запятой		● <sup>1</sup>		●	●	—
Строковые величины		● <sup>2</sup>		—	—	—

<sup>1</sup> кроме центральных процессоров Q00J, Q00 и Q01

<sup>2</sup> В центральных процессорах Q00J, Q00 и Q01 строковые величины можно использовать только в сочетании с командой \$MOV.

**ПРИМЕЧАНИЕ** Более подробную информацию на эту тему вы найдете в разд. 3.5 „Типы данных“ этого руководства.

### А.3.4 Сравнение таймеров

#### Функции таймеров

Название	Функция		Q QnA/QnAR	AnU *	AnA *	AnN/AnS*	
Низкоскоростные таймеры	Измерительный диапазон		100 мс (предварительная настройка) Это значение можно регулировать в параметрах в диапазоне от 10 до 100 мс. Это значение служит в качестве множителя для заданного значения (TV).	(неизменная настройка на 100 мс)			
	Программирование (GX IEC Developer)	TIMER_M (обычные/расширенные таймеры)	передача заданного значения и вызов (запуск) таймера	●	●	●	●
		TIMER_VALUE_M (только расширенные таймеры)	передача заданного значения	●	●	●	●
		TIMER_START_M (только расширенные таймеры)	вызов (запуск) таймера	●	●	●	●
	Программирование (GX Developer)	передача заданного значения и вызов (запуск) таймера	OUT Tn заданное значение	OUT Tn заданное значение			
Высокоскоростные таймеры	Измерительный диапазон		10 мс (предварительная настройка) Это значение можно регулировать в параметрах в диапазоне от 10 до 100 мс. Это значение служит в качестве множителя для заданного значения (TV).	● неизменная настройка на 10 мс			
	Программирование (GX IEC Developer)	TIMER_M (обычные/расширенные таймеры)	передача заданного значения и вызов (запуск) таймера	●	●	●	●
		TIMER_VALUE_M (только расшир. таймеры)	передача заданного значения	●	●	●	●
		TIMER_START_M (только расшир. таймеры)	вызов (запуск) таймера	●	●	●	●
	Программирование (GX Developer)	передача заданного значения и вызов (запуск) таймера	OUTH Tn заданное значение	OUTH Tn заданное значение			



Функции таймеров

Название	Функция		Q QnA/QnAR	AnU *	AnA *	AnN/AnS*	
Фиксируемый низкоскоростной таймер	Измерительный диапазон		100 мс (предвар. настройка) Это значение можно регулировать в параметрах в диапазоне от 10 до 100 мс. Это значение является коэффициентом, на который умножается заданн. знач. (TV).	● (неизменная настройка на 100 мс)			
	Программирование (GX IEC Developer)	TIMER_H_M (обычные/расшир. таймеры)	передача заданного значения и вызов (запуск) таймера	●	●	●	●
		TIMER_VALUE_M (только расшир. таймеры)	передача заданного значения	●	●	●	●
		TIMER_START_M (только расшир. таймеры)	вызов (запуск) таймера	●	●	●	●
Программирование (GX Developer)	передача заданного значения и вызов (запуск) таймера	OUT STn заданное значение	OUT Tn заданное значение				
Фиксируемый высокоскоростной таймер	измерительный диапазон		10 мс (предвар. настройка) Это значение можно регулировать в параметрах в диапазоне от 1 до 100 мс. Это значение служит в качестве коэффициента, на который умножается заданное значение (TV).	—			
	Программирование (GX IEC Developer)	TIMER_H_M (обычные/расширенные таймеры)	передача заданного значения и вызов (запуск) таймера	●	—	—	—
		TIMER_VALUE_M (только расшир. таймеры)	передача заданного значения	●	—	—	—
		TIMER_START_M (только расшир. таймеры)	вызов (запуск) таймера	●	—	—	—
Программирование (GX Developer)	передача заданного значения и вызов (запуск) таймера	OUTH STn заданное значение	—	—	—		
Диапазон настройки заданного значения			от 1 до 32767	от 1 до 32767			
Обработка заданного значения 0			немедленное включение	без максимума (не выдает время)			
Индексируемая адресация	контакт		возможна (можно использовать только Z0 и Z1)	возможна		не возможна	
	катушка			не возможна		не возможна	
	заданное значение		не возможна	не возможна		не возможна	
	фактическое значение		возможна (можно использовать Z0 ... Z15; в процессорах Q00J, Q00 и Q01 – только Z0 ... Z9)	возможна		возможна	
Обновление фактического значения			при выполнении команды OUT Tn	после обработки команды END			
Обработка включения/выкл. контакта							

\* В GX IEC Developer начальный адрес различных таймеров должен быть установлен в диалоговом окне "Область таймеров/счетчиков".

## Функциональные блоки таймеров в GX IEC Developer

Название	Функциональный блок	Тип центрального процессора					
		Q	QnA/QnAR	AnU	AnA	AnN	AnS
10-миллисекундный таймер	Instance TIMER_10_FB_M - Coil ValueOut - Preset Status - ValueIn	—	●	●	●	●	●
100-миллисекундный таймер	Instance TIMER_100_FB_M - Coil ValueOut - Preset Status - ValueIn	—	●	●	●	●	●
Фиксируемый таймер	Instance TIMER_CONT_FB_M - Coil ValueOut - Preset Status - ValueIn	—	●	●	●	●	●
Низкоскоростной таймер	Instance TIMER_LOW_FB_M - Coil ValueOut - Preset Status - ValueIn	—	●	—	—	—	—
Высокоскоростной таймер	Instance TIMER_HIGH_FB_M - Coil ValueOut - Preset Status - ValueIn	—	●	—	—	—	—
Фиксируемый высокоскоростной таймер	Instance TIMER_CONTHFB_M - Coil ValueOut - Preset Status - ValueIn	—	●	—	—	—	—

## Функциональные блоки таймеров (разъяснение)

Термин в функциональном блоке	Значение		Обозначение для обычных таймеров	Обозначение для фиксируемых таймеров
	значение	значение		
Coil	катушка	условие включения процесса таймера	TC	STC
Preset	заданное значение		TValue	TValue
ValueIn	начальное значение	обычно: 0	—	—
ValueOut	фактическое значение		TN	STN
Status	контакт	выходной контакт переключается после отработки	TS	STS

Пометьте функциональный блок установленным в заголовке описателем экземпляра и присвойте входные и выходные переменные.

### Указания по применению таймеров

При выполнении команды OUT(H) T обновляется настройка таймера и устанавливается или сбрасывается катушка таймера. Если фактическое значение таймера больше или равно заданному значению, катушка таймера устанавливается.

В программе, в которой таймер запускается другим таймером, необходимо, чтобы сначала обрабатывались команды для таймера, запускаемого позднее. Например, если катушка таймера T1 запускает таймер T2, то команды, относящиеся ко времени таймера T2, в программе были расположены перед аналогичными командами для T1.

Так предотвращается установка всех катушек таймеров в одном и том же цикле. Это происходит, если у быстрых таймеров заданное значение меньше времени цикла или если у медленных таймеров заданное значение установлено на "1" и команды второго таймера обрабатываются после запускающего таймера.

### А.3.5 Сравнение счетчиков

#### Функции счетчиков

Функция			Тип центрального процессора					
			Q	QnA/ QnAR	AnU	AnA	AnN	AnS
Программирование (GX IEC Developer)	Counter_M	передача заданного значения и вызов (запуск) счетчика	●		●	●	●	●
	Counter_Start_M	передача заданного значения	●		●	●	●	●
	Counter_Value_M	вызов (запуск) счетчика	●		●	●	●	●
Программирование (GX Developer)	OUT Sp заданное значение	передача заданного значения и вызов (запуск) счетчика	●		●	●	●	●
Индексная адресация	контакт		возможна (можно использовать только Z0 и Z1)		возможна		не возможна	
	катушка		возможна (можно использовать только Z0 и Z1)		возможна		не возможна	
	заданное значение		не возможна		не возможна		не возможна	
	фактическое значение		возможна (можно использовать Z0 ... Z15; в процессорах Q00J, Q00 и Q01 – только Z0 ... Z9)		возможна		возможна	
Обновление фактического значения			при выполнении команды OUT Sp		после обработки команды END			
Обработка включения/выключения контакта								

## Функциональные блоки счетчиков

Название	Функциональный блок	Тип центрального процессора					
		Q	QnA/QnAR	AnU	AnA	AnN	AnS
Счетчик		—	●	●	●	●	●

## Функциональные блоки счетчиков (разъяснение)

Термин в функциональном блоке	Значение		Обозначение счетчика
Coil	катушка	условие включения процесса счетчика	CC
Preset	заданное значение		CValue
ValueIn	начальное значение	обычно: 0	—
ValueOut	фактическое значение		CN
Status	контакт	выходной контакт переключается после отработки функционального блока	CS

## А.3.6 Сравнение команд индикации

Команда	Q	QnA/QnAR	AnU	AnA	AnN	AnS
PR <sup>1</sup>	SM701 не установлен: вывод происходит, пока не будет достигнут код 00 <sub>H</sub> . SM701 установлен: вывод 16 знаков.		M9049 не установлен: вывод происходит, пока не будет достигнут код 00 <sub>H</sub> . M9049 установлен: вывод 16 знаков			
PRC <sup>1</sup>	SM701 не установлен: вывод 32 знаков комментария SM701 установлен: вывод 16 старших разрядов		вывод 16 знаков комментария			

<sup>1</sup> кроме центральных процессоров Q00J, Q00 и Q01

### А.3.7 Команды серий "Q" и "System Q", эквивалентные командам MELSEC серии "A"

Так как центральные процессоры Q/QnA не используют сумматоры (A0, A1), команды для процессоров AnU, AnN и AnN, использующие сумматоры, имеют иной формат.

Функция	Процессоры Q/QnA		Процессоры AnU/AnA/AnN	
	Формат команды	Примечание	Формат команды	Примечание
16-битное вращение битов вправо	ROR (d, n)	d: вращаемые данные	ROR (n)	Вращаемые данные установлены в A0.
	RCR (d, n)	d: вращаемые данные В качестве флага переноса используется SM700.	RCR (n)	Вращаемые данные установлены в A0. В качестве флага переноса используется M9012.
16-битное вращение битов влево	ROL (d, n)	d: вращаемые данные	ROL (n)	Вращаемые данные установлены в A0.
	RCL (d, n)	d: вращаемые данные В качестве флага переноса используется SM700.	RCL (n)	Вращаемые данные установлены в A0. В качестве флага переноса используется M9012.
32-битное вращение битов вправо	DROR (d, n)	d: вращаемые данные	DROR (n)	Вращаемые данные установлены в A0 и A1.
	DRCR (d, n)	d: вращаемые данные В качестве флага переноса используется SM700.	DRCR (n)	Вращаемые данные установлены в A0 и A1. В качестве флага переноса используется M9012.
32-битное вращение битов влево	DROL (d, n)	d: вращаемые данные	DROL (n)	Вращаемые данные установлены в A0 и A1.
	DRCL (d, n)	d: вращаемые данные В качестве флага переноса используется SM700.	DRCL (n)	Вращаемые данные установлены в A0 и A1. В качестве флага переноса используется M9012.
Поиск 16-битных данных	SER (s1, s2, d, n)	Результат поиска сохраняется в операндах, указанных в D и D + 1.	SER (s1, s2, n)	Результат поиска сохраняется в A0 и A1.
Поиск 32-битных данных	DSER (s1, s2, d, n)	Результат поиска сохраняется в операндах, указанных в D и D + 1.	DSER (s1, s2, n)	Результат поиска сохраняется в A0 и A1. В качестве флага переноса используется M9012.
16-битный контроль битов данных	SUM (s, d)	Результат контроля сохраняется в операнде, указанном в D.	SUM (s)	Результат контроля сохраняется в A0.
32-битный контроль битов данных	DSUM (s, d)	Результат контроля сохраняется в операнде, указанном в D.	DSUM (s)	Результат контроля сохраняется в A0.
Частичное обновление	RFS (s, n)	Добавлена специализированная команда.	SEG (d, n)	Только если установлен M9052.
8 знаков Преобразование в ASCII	\$MOV (s, d)		ASC (d)	
Установка флага переноса	SET (SM700)	Без специализированной команды	STC	
Сброс флага переноса	RST (SM700)	Без специализированной команды	CLC	
Переход к концу программы	GOEND	Добавлена специализированная команда.	CJ (P255)	P255: обозначение команды End
Команда CHK <sup>1</sup>	CHKST CHK	Добавлена команда CHKST	CJ (Pn) CHK (P255)	

<sup>1</sup> кроме центральных процессоров Q00J, Q00 и Q01

### А.3.8 Сравнение процессоров QnA/Q2AS с процессорами MELSEC System Q

Следующие команды для процессоров Q серии MELSEC "System Q" являются новыми по отношению к сериям MELSEC QnA/Q2AS.

Функция	Команда
Считывание информации модуля	UNIRD
Установка трассировки	TRACE
Сброс трассировки	TRACER
Запись данных в файл	S.FWRITE
Считывание данных из файла	S.FREAD
Загрузка программы с карты памяти	PLOAD
Стирание программы	PUNLOAD
Стирание программы и загрузка программы с карты памяти	PSWAP
Передача блоков двоичных данных	PBMOV
Запись в общую область памяти (только процессоры Q02, Q02H, Q06H, Q12H и Q025, начиная с версии "B")	S.TO

В центральных процессорах "System Q" более не имеется команд, указанных в следующей таблице.

Функция	Команда
Запись данных в регистр EEPROM	EROMWR
Установка семплирования (выборочной трассировки) (вместо этого можно использовать TRACE)	STRA
Сброс семплирования (выборочной трассировки) (вместо этого можно использовать TRACER)	STRAR
Установка фиксации состояния	SLT
Сброс фиксации состояния	SLTR
Установка контроля (трассировки) программы	PTRA
Сброс контроля (трассировки) программы	PTRAR
Выполнение контроля (трассировки) программы	PTRAEXE PTRAEHEP
Вывод знаков ASCII на светодиодный дисплей	LED
Вывод комментариев на светодиодный дисплей	LEDC

В отношении следующих команд следует учитывать различия между процессорами QnA/Q2AS и процессорами MELSEC System Q.

Функция	Команда
Установка выходов, установка/сброс операнда (только внутреннего)	OUT, SET, RST
Считывание комментариев	COMRD
Вывод кодов ASCII	PRC
Стирание индикации	LEDR
Конвертирование в формат BIN	BIN
Конвертирование в формат DBIN	DBIN
Считывание данных времени	DATERD
Запись данных времени	DATEWR
Битовые схемы условий выполнения программ прерывания	IMASK
Команда обновления сетевых и интерфейсных данных	COM
Обновление сетевых данных	ZCOM
Считывание информации маршрутизации	RTREAD
Запись информации маршрутизации	RTWRITE
Настройка ПИД-регулирования	PIDINT
ПИД-регулирование	PIDCONT
Однофазный возрастающий/убывающий счетчик	UDCNT1
Двухфазный возрастающий/убывающий счетчик	UDCNT2
Счетчик импульсов	SPD
Импульсный выход с регулируемым количеством импульсов	PLSY
Широтно-импульсная модуляция	PWM

#### ПРИМЕЧАНИЯ

Если программа для процессоров QnA, обращающаяся к специальным модулям, конвертируется в программу для процессора "System Q", то при этом необходимо учитывать следующие пункты:

- В режиме "Q" модули центральных процессоров "System Q" не совместимы со специальными модулями и модулями сетевой коммуникации серий A/AnS. Если вы продолжаете использовать эти модули, измените программы, применив в них команды FROM и TO.
- Если вы заменяете специальные модули серий QnA, Q2AS, A или AnS специальными модулями System Q, то некоторые команды можно продолжать использовать. Более подробную информацию на эту тему можно найти в руководствах по соответствующим специальным модулям.

## A.4 Обзор специальных маркеров

### A.4.1 Перечень маркеров диагностики (серии "MELSEC Q" и "MELSEC System Q")

Маркеры диагностики (SM) являются внутренними маркерами, применение которых в контроллере жестко установлено. Поэтому их нельзя использовать в программах аналогично внутренним маркерам. Однако их можно включать и выключать для управления центральным процессором.

**ПРИМЕЧАНИЯ** Маркеры диагностики SM1200 ... SM1255 используются в процессорах QnA. В процессорах серии "Q" эти маркеры не используются.

Маркеры диагностики, начиная с SM 1500, зарезервированы для процессора Q4AR.

В этой таблице разъясняются заголовки столбцов, используемые в таблицах на следующих страницах.

Заголовок таблицы	Значение
Адрес	● Показывает адрес маркера диагностики.
Название	● Показывает название маркера диагностики.
Значение	● Краткое разъяснение значения маркера диагностики.
Описание	● Подробная информация о значении маркера диагностики.
Устанавливает (если установлен)	<p>Маркер диагностики может устанавливаться системой или пользователем.</p> <p>&lt;Устанавливает&gt;</p> <p>С : устанавливает система</p> <p>П : устанавливает пользователь (в основной программе или в тестовом режиме периферийного устройства)</p> <p>С/П : устанавливает система и пользователь</p> <p>Показано только в том случае, если установка была выполнена системой.</p> <p>&lt;Если установлен&gt;</p> <p>Обработка команды END : устанавливается при каждой обработке команды END</p> <p>Инициализация : устанавливается только во время инициализации (при включении блока питания или при переключении центрального процессора из режима "STOP" в режим "RUN")</p> <p>Изменение состояния : устанавливается только после изменения состояния</p> <p>Ошибка : устанавливается только после возникновения ошибки</p> <p>Выполнение команды : устанавливается при выполнении команды</p> <p>Запрос : устанавливается только в том случае, если имеется запрос пользователя (сделанный с помощью маркера SM или т. п.)</p>
A-CPU M9[ ][ ][ ]	<ul style="list-style-type: none"> <li>● [ ][ ][ ], соответствующий процессору "A". (изменения и иное написание, если оно также изменяется)</li> <li>● Если в процессоре "Q" он добавлен впервые, то в таблице он обозначен словом "новый".</li> </ul>
Действ. для:	<ul style="list-style-type: none"> <li>● Указывает, для какого центрального процессора предназначен этот специальный маркер.</li> <li>U: действителен для центральных процессоров всех типов</li> <li>Q: действителен только для всех модулей центральных процессоров серии "System Q"</li> <li>QnA: действителен для центральных процессоров серий QnA и Q2AS</li> <li>Тип центрального процессора: действителен только для данного центрального процессора (например, Q4AR)</li> <li>Rem: действителен для удаленных модулей ввода-вывода MELSECNET/H</li> </ul>



(1) Информация для диагностики ошибок

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" М9[ ][ ][ ]	Дейст. для:
SM0	Ошибка, выявленная при диагностике	выкл.: ошибок нет вкл.: ошибка	Устанавливается в состояние "включено", если результат диагностики показывает ошибку (включая внешнюю диагностику).  После устранения ошибки маркер остается установленным.	С (ошибка)	новый	● Rem
SM1	Ошибка, выявленная путем самодиагностики	выкл.: при самодиагностике ошибок не выявлено вкл.: ошибка	Устанавливается в состояние "включено", если результат самодиагностики показывает ошибку.  После устранения ошибки маркер остается установленным.	С (ошибка)	M9008	
SM5	Общая информация об ошибке	выкл.: общей информации об ошибке нет вкл.: общая информация об ошибке	При установленном SM0 и наличии общей информации об ошибке устанавливается в состояние "включено"	С (ошибка)	новый	
SM16	Специальная информация об ошибке	выкл.: специальной информации об ошибке нет вкл.: специальная информация об ошибке	При установленном SM0 и наличии специальной информации об ошибке устанавливается в состояние "включено"	С (ошибка)	новый	
SM50	Сброс ошибки	выкл. → вкл.: стирание ошибки	Ошибка сбрасывается.  Дополнительная информация имеется в разделе 5.3.6.	П	новый	
SM51	Низкое напряжение батареи (фиксируемый маркер)	выкл.: нормальное напряжение вкл.: напряжение упало	Напряжение батареи буферного питания центрального процессора или карты памяти снизилось ниже минимального предела.  После замены батареи маркер остается установленным.  Состояние маркера совпадает с состоянием светодиода "BAT. ALARM".	С (ошибка)	M9007	●
SM52	Низкое напряжение батареи	выкл.: нормальное напряжение вкл.: напряжение упало	Напряжение батареи буферного питания снизилось ниже минимального предела.  После замены батареи маркер сбрасывается.	С (ошибка)	M9006	
SM53	Падение напряжения питания	выкл.: нормальное напряжение вкл.: напряжение упало	Входное напряжение сетевого блока переменного напряжения исчезло на время менее 20 мс. Сброс происходит при выключении и повторном включении напряжения питания.	С (ошибка)	M9005	●
			Входное напряжение блока питания с входом постоянного напряжения исчезло на время менее 10 мс. Сброс происходит при выключении и повторном включении напряжения питания.			Q
			Входное напряжение блока питания с входом постоянного напряжения исчезло на время менее 1 мс. Сброс происходит при выключении и повторном включении напряжения питания.			QnA
SM54	Ошибка в MELSECNET/ MINI	выкл.: нормальное состояние вкл.: ошибка	Этот маркер устанавливается при возникновении ошибки связи в установленном модуле AJ71PT32 (S3).  Маркер остается установленным и после исчезновения ошибки.	С (ошибка)	M9004	QnA
SM56	Ошибка обработки	выкл.: нормальное состояние вкл.: ошибка обработки	Этот маркер устанавливается при возникновении ошибки обработки.  Маркер остается установленным и после исчезновения ошибки.	С (ошибка)	M9011	●

(1) Информация для диагностики ошибок

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM60	Неисправен предохранитель	выкл.: нормальное состояние вкл.: модуль с неисправным предохранителем	Этот маркер устанавливается при обнаружении неисправности предохранителя в одном из выходных модулей.  Маркер остается установленным и после возврата в нормальное состояние.	С (ошибка)	M9000	● Rem
SM61	Ошибка, выявленная при сверке модулей ввода-вывода	выкл.: нормальное состояние вкл.: при сверке выявлена ошибка	Текущее состояние модулей ввода-вывода отличается от зарегистрированной информации после включения напряжения питания. Сверка модулей ввода-вывода выполняется и для удаленной станции.	С (ошибка)	M9002	
SM62	Индикатор маркера ошибки	выкл.: не распознан вкл.: распознан	Устанавливается, если установлен только один маркер ошибки F.	С (выполнение команды)	M9009	●
SM80	Ошибка, выявленная с помощью команды CHK	выкл.: нормальное состояние вкл.: ошибка	Устанавливается при обнаружении ошибки с помощью команды CHK. Маркер остается установленным и после исчезновения ошибки.	С (выполнение команды)	новый	
SM90	Запуск WDT (контрольного таймера) для контроля переходов  (действует только при наличии программы на языке SFC)	выкл.: не запущен (WDT сброшен)  вкл.: запуск (WD запускается)	соответствует SD90	П  Этот маркер устанавливается при начале измерения с помощью WDT. При сбросе маркера таймер WDT сбрасывается.	M9108	QnA, Q, кроме Q001, Q00 и Q01
SM91			соответствует SD91		M9109	
SM92			соответствует SD92		M9110	
SM93			соответствует SD93		M9111	
SM94			соответствует SD94		M9112	
SM95			соответствует SD95		M9113	
SM96			соответствует SD96		M9114	
SM97			соответствует SD97		новый	
SM98			соответствует SD98		новый	
SM99			соответствует SD99		новый	
SM100	Применение последовательной коммуникации	выкл.: последовательная коммуникация не используется вкл.: последовательная коммуникация используется	Этот маркер показывает, выбрана ли последовательная коммуникация в системных настройках.	С (инициализация)	новый	
SM101	Протокол, применяемый при последовательной коммуникации	выкл.: протокол для программаторов вкл.: протокол MC	Этот маркер показывает, по какому протоколу происходит последовательная коммуникация.	С (при последовательной коммуникации)	новый	
SM110	Ошибка протокола	выкл.: ошибок нет вкл.: ошибка	Устанавливается при попытке обмена данными через последовательный интерфейс на основе неправильного протокола.  Остается установленным, даже если после этого для коммуникации используется правильный протокол.	С (ошибка)	новый	Q001, Q00 и Q01
SM111	Состояние коммуникации	выкл.: ошибок нет вкл.: ошибка	Устанавливается, если действительно используемый тип последовательной коммуникации отличается от настроек.  Остается установленным, даже если после этого сообщения коммуникация ведется в правильном режиме.	С (ошибка)	новый	
SM112	Стирание информации об ошибках	вкл.: стирание маркеров диагностики и регистров	При переходе из состояния "выкл." в состояние "вкл." стираются маркеры SM110, SM111, SD110 и SD111.	П	новый	

## (1) Информация для диагностики ошибок

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" М9[ ][ ][ ]	Дейст. для:
SM113	Переполнение данных	выкл.: ошибок нет вкл.: ошибка	Этот маркер устанавливается, если при последовательной коммуникации произошло переполнение данных.	С (ошибка)	новый	Q00J, Q00 и Q01
SM114	Ошибка четности	выкл.: ошибок нет вкл.: ошибка	Установленный маркер показывает ошибку четности при последовательной коммуникации.	С (ошибка)	новый	
SM115	Ошибка кадра	выкл.: ошибок нет вкл.: ошибка	Этот маркер устанавливается, если при последовательной коммуникации имеется ошибка кадра данных.	С (ошибка)	новый	

## (2) Системная информация

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" М9[ ][ ][ ]	Дейст. для:
SM202	Команда выключения светодиода	выкл. → вкл.: выключение светодиода	Светодиоды, сопоставленные битам SD202, гаснут при изменении состояния битов с "выкл." на "вкл."	П	новый	● кроме Q00J, Q00 и Q01
SM203	Маркер состояния "STOP"	Состояние "STOP"	Устанавливается при останове центрального процессора.	С (изменение состояния)	M9042	●
SM204	Маркер состояния "PAUSE"	Состояние "PAUSE"	Устанавливается, если центральный процессор находится в режиме "Пауза".	С (изменение состояния)	M9041	
SM205	Маркер режима "STEP-RUN"	Режим "STEP-RUN"	Устанавливается, если центральный процессор находится в режиме "STEP-RUN".	С (изменение состояния)	M9054	● кроме Q00J, Q00 и Q01
SM206	Условие выполнения для состояния "PAUSE"	выкл.: состояние не возможно вкл.: состояние возможно	Центральный процессор переходит в состояние "PAUSE", если установлены дистанционный контакт "PAUSE" и маркер.	П	M9040	●
	Состояние тестирования операнда	выкл.: тестирование операнда еще не выполнено вкл.: тестирование операнда выполнено	Этот маркер указывает состояние тестирования операнда, которое можно выполнить с помощью среды программирования.	С (запрос)	новый	Q00J, Q00 и Q01 Rem
SM210	Запрос на установку данных часов	выкл.: не обрабатывается вкл.: запрос	При установленном маркере данные часов после выполнения команды END сохраняются в регистрах SD210 ... SD213 и передаются в часы.	П	M9025	●
SM211	Ошибка данных часов	выкл.: нормальное состояние вкл.: ошибка	Маркер установлен, если в данных часов, сохраненных в регистрах с SD210 по SD213, имеется ошибка. Если ошибок нет, маркер не установлен.	С (запрос)	M9026	
SM212	Индикация данных часов	выкл.: не обрабатывается вкл.: индикация	Данные часов из регистров с SD210 по SD213 считываются и выводятся на светодиодный индикатор центрального процессора с индикацией месяца, дня, часа, минуты и секунды. (это возможно только в центральных процессорах Q3A и Q4A)	П	M9027	Q3A, Q4A, Q4AR
SM213	Запрос на считывание данных часов	выкл.: не обрабатывается вкл.: запрос	При установленном маркере данные часов считываются в регистры SD210...SD213 в виде двоично-десятичных значений.	П	M9028	● Rem

## (2) Системная информация

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM240	Маркер сброса центрального процессора 1	выкл.: сброса нет вкл.: на модуле центр. процессора 1 выполнен сброс	Этот маркер устанавливается при сбросе центрального процессора 1 или при отсоединении центрального процессора от монтажной шины. Другие центральные процессоры многопроцессорной системы также сбрасываются.	С (изменение состояния)	новый	Q02, Q02H, Q06H, Q12H, Q25H, начиная с версий "В"
SM241	Маркер сброса центрального процессора 2	выкл.: сброса нет вкл.: на модуле центр. процессора 2 выполнен сброс	Этот маркер устанавливается при сбросе центрального процессора 2 или при отсоединении центрального процессора от монтажной шины. В других процессорах многопроцессорной системы выводится сообщение об ошибке MULTI CPU DOWN (код ошибки 7000).	С (изменение состояния)	новый	
SM242	Маркер сброса центрального процессора 3	выкл.: без сброса вкл.: на модуле центр. процессора 3 выполнен сброс	Этот маркер устанавливается при сбросе центрального процессора 3 или при отсоединении центрального процессора от монтажной шины. В других процессорах многопроцессорной системы выводится сообщение об ошибке MULTI CPU DOWN (код ошибки 7000).	С (изменение состояния)	новый	
SM243	Маркер сброса центрального процессора 4	выкл.: сброса нет вкл.: на модуле центр. процессора 4 выполнен сброс	Этот маркер устанавливается при сбросе центрального процессора 4 или при отсоединении центрального процессора от монтажной шины. В других процессорах многопроцессорной системы выводится сообщение об ошибке MULTI CPU DOWN (код ошибки 7000).	С (изменение состояния)	новый	
SM244	Маркер ошибки центрального процессора 1	выкл.: ошибок нет вкл.: ошибка в центр. процессоре 1, останавливающая центр. процессор	Установленный маркер показывает, что возникла ошибка, остановившая центральный процессор. Если ошибок нет либо имеется ошибка, не вызывающая перехода процессора в состояние "STOP", этот маркер сбрасывается.	С (изменение состояния)	новый	
SM245	Маркер ошибки центрального процессора 2	выкл.: ошибок нет вкл.: ошибка в центр. процессоре 2, останавливающая центр. процессор		С (изменение состояния)	новый	
SM246	Маркер ошибки центрального процессора 3	выкл.: ошибок нет вкл.: ошибка в центр. процессоре 3, останавливающая центр. процессор		С (изменение состояния)	новый	
SM247	Маркер ошибки центрального процессора 4	выкл.: ошибок нет вкл.: ошибка в центр. процессоре 4, останавливающая центр. процессор		С (изменение состояния)	новый	
SM250	Считывание макс. загруженного кол. входов и выходов	выкл.: не обрабатывается вкл.: считывание	Этот маркер устанавливается при считывании максимального загруженного количества входов и выходов в регистр SD250.	П	новый	● кроме Q00J, Q00 и Q01Rem
SM251	Маркер изменения модулей ввода-вывода	выкл.: изменения нет вкл.: изменение	После задания головного адреса для определенного модуля ввода-вывода в регистре SD251 этот модуль можно переключить онлайн путем установки регистра SD251 (только один модуль при каждом выполнении). Изменить рабочее состояние модуля ввода-вывода с помощью маркера SD251 можно и во время тестирования с помощью программатора или при остановленном состоянии центрального процессора. До завершения переключения модуля ввода-вывода в режим онлайн нельзя изменить рабочее состояние центрального процессора.	П (обработка команды END)	M9054	Q2A(S1), Q3A, Q4A, Q4AR
SM252	Замена ввода-вывода разрешена	выкл.: замена заблокирована вкл.: замена возможна	Устанавливается, если замена ввода-вывода разрешена.	С (обработка команды END)	новый	
SM254	Команда обновления для всех станций	выкл.: обновл. только для 1-й станции вкл.: обновление для всех станций	Действительно для пакетного обновления и для медленного цикла. Этот маркер устанавливается, если обновление должно происходить для всех станций.	С (обработка команды END)	новый	Q, кроме Q00J, Q00 и Q01

## (2) Системная информация

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Действ. для:
SM255	Информация модуля 1 MELSECNET/10	выкл.: активная сеть вкл.: резервная сеть	Устанавливается в случае резервной сети. (Если определенное состояние (резервное или активное) не указано, предполагается активное состояние)	С (инициализация)	новый	● кроме Q00J, Q00 и Q01
SM256		выкл.: считывание происходит вкл.: считывание не происходит	Для обновления данных, обмениваемых между модулем связи и центральным процессором (В, W и т. д.), указывается, происходит ли считывание из модуля связи.	П	новый	
SM257		выкл.: запись происходит вкл.: запись не происходит	Для обновления данных, обмениваемых между центральным процессором и модулем связи (В, W и т. п.), указывается, происходит ли запись в модуль связи.	П	новый	
SM260	Информация модуля 2 MELSECNET/10	выкл.: активная сеть вкл.: резервная сеть	Устанавливается в случае резервной сети. (Если определенное состояние (резервное или активное) не указано, предполагается активное состояние)	С (инициализация)	новый	
SM261		выкл.: считывание происходит вкл.: считывание не происходит	Для обновления данных, обмениваемых между модулем связи и центральным процессором (В, W и т. д.), указывается, происходит ли считывание из модуля связи.	П	новый	
SM262		выкл.: запись происходит вкл.: запись не происходит	Для обновления данных, обмениваемых между центральным процессором и модулем связи (В, W и т. п.) указывается, происходит ли запись в модуль связи.	П	новый	
SM265	Информация модуля 3 MELSECNET/10	выкл.: активная сеть вкл.: резервная сеть	Устанавливается в случае резервной сети. (Если определенное состояние (резервное или активное) не указано, предполагается активное состояние)	С (инициализация)	новый	
SM266		выкл.: считывание происходит вкл.: считывание не происходит	Для обновления данных, обмениваемых между модулем связи и центральным процессором (В, W и т. д.), указывается, происходит ли считывание из модуля связи.	П	новый	
SM267		выкл.: запись происходит вкл.: запись не происходит	Для обновления данных, обмениваемых между центральным процессором и модулем связи (В, W и т. п.), указывается, происходит ли запись в модуль связи.	П	новый	
SM270	Информация модуля 4 MELSECNET/10	выкл.: активная сеть вкл.: резервная сеть	Устанавливается в случае резервной сети. (Если определенное состояние (резервное или активное) не указано, предполагается активное состояние.)	С (инициализация)	новый	
SM271		выкл.: считывание происходит вкл.: считывание не происходит	Для обновления данных, обмениваемых между модулем связи и центральным процессором (В, W и т. д.), указывается, происходит ли считывание из модуля связи.	П	новый	
SM272		выкл.: запись происходит вкл.: запись не происходит	Для обновления данных, обмениваемых между центральным процессором и модулем связи (В, W и т. п.), указывается, происходит ли запись в модуль связи.	П	новый	
SM280	Ошибка CC-Link	выкл.: нормальное состояние вкл.: ошибка	Устанавливается, если в одном из установленных модулей QJ61QBT11 CC-Link распознана ошибка. Если снова восстановлено нормальное рабочее состояние, маркер сбрасывается.	С (изменение состояния)	новый	Q, кроме Q00J, Q00 и Q01
			Устанавливается, если в одном из установленных модулей A1 (1S)J61QBT11 CC-Link распознана ошибка. При восстановлении нормального рабочего состояния остается установленным.	С (ошибка)	новый	QnA
SM315	Деблокировка времени, зарезервированного для коммуникации	выкл.: без задержки вкл.: с задержкой	При установленном маркере, если коммуникация не происходит, выполнение команды END задерживается на время, указанное в SD315. Время цикла соответственно удлиняется. При сброшенном маркере, если коммуникация не происходит, команда END выполняется без задержки.	П	новый	Q00J, Q00 и Q01

## (2) Системная информация

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM320	Программа на языке SFC имеется	выкл.: программы языка SFC не имеется вкл.: программа на языке SFC имеется	Маркер установлен, если программа на языке SFC правильно зарегистрирована, и не установлен, если она не зарегистрирована. Если специализированная команда на языке SFC содержит ошибку, маркер не устанавливается.	С (инициализация)	M9100	
SM321	Запуск/останов программы на языке SFC	выкл.: программа на языке SFC останавливается вкл.: программа на языке SFC запускается	Значение инициализации устанавливается на такое же значение, как маркер SM900. (Если имеется программа на языке SFC, маркер устанавливается автоматически.) Если перед выполнением программы на языке SFC маркер находится в выключенном состоянии, программа не выполняется. Если после этого состояние маркера изменилось с выключенного на включенное, программа на языке SFC запускается. Если затем состояние маркера изменилось с включенного состояния на выключенное, программа на языке SFC останавливается.	С/П (инициализация)	M9101 изменился формат	
SM322	Пусковое состояние программы на языке SFC	выкл.: первоначальный запуск вкл.: перезапуск	В зависимости от параметров, начальным состоянием является либо включенное, либо выключенное. Если этот маркер выключен, то все состояния выполнения с момента остановки программы на языке SFC сбрасываются. Запуск осуществляется с шага инициализации, из которого был сделан запрос запуска. Если этот маркер включен, программа перезапускается с того места, которое было активным при останове. (Включенное состояние возможно только в том случае, если в параметрах было выбрано возобновление программы.) Для SM902 не предусмотрена автоматическая фиксация.	С/П (инициализация)	M9102 изменился формат	
SM323	Наличие/отсутствие непрерывных переходов во всем блоке	выкл.: переход не действует вкл.: переход действует	Если маркер не установлен, в каждом цикле обрабатывается один из имеющихся переходов. Если маркер установлен, в каждом цикле обрабатываются все непрерывные переходы. Чтобы различать отдельные блоки, бит непрерывного перехода в блоке получает приоритет. (Различие проверяется перед запуском блока.)	П	M9103	● кроме Q00J, Q00 и Q01
SM324	Маркер предотвращения непрерывного перехода	выкл.: переход выполнен вкл.: переход выполняется	Маркер устанавливается при выполнении действующего непрерывного перехода. По окончании выполнения маркер сбрасывается. В нормальном состоянии маркер устанавливается, если переход не действует.	С (выполнение команды)	M9104	
SM325	Режим вывода при останове блока	выкл.: выключено вкл.: сохранение состояния	При останове блока выводится активный шаг обработки. При выключенном маркере все выходы сбрасываются. При включенном маркере все выходы сохраняют свое состояние.	С (изменение состояния)	M9196	
SM326	Режим вывода для программы на языке SFC	выкл.: стирание выходов вкл.: сохранение состояния	С помощью этого маркера выбирается реакция выходов после изменения имеющейся программы SFC и запуска центрального процессора.	П	новый	
SM327	Режим вывода при выполнении команды End	выкл.: выключено вкл.: сохранение состояния	С помощью этого маркера выбирается, должны ли выходы сохранять свое состояние при выполнении последнего шага программы. Если маркер не установлен, все выходы сбрасываются. При установленном маркере выходы сохраняют свое состояние.	С (инициализация) П	новый	
SM330	Режим для программ с медленным выполнением	выкл.: асинхронный вкл.: синхронный	Асинхронный: команды для программы с медленным выполнением продолжают выполняться на протяжении времени, еще имеющегося в цикле. Синхронный: команды для программы с медленным выполнением начинают выполняться в следующем цикле, даже если еще остается достаточно времени.	П (обработка команды END)	новый	

(3) Системные такты и счетчики

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM400	Всегда включен	Вкл. Выкл.	Этот специальный маркер установлен (включен) всегда.	С (обработка команды END)	M9036	
SM401	Всегда выключен	Вкл. Выкл.	Этот специальный маркер сброшен (выключен) всегда.	С (обработка команды END)	M9037	
SM402	Включен только на один программный цикл после RUN	Вкл. Выкл.	После установки режима RUN программа на один программный цикл устанавливается в состояние "включено". Этот способ могут использовать только программы, выполняемые один раз за цикл.	С (обработка команды END)	M9038	●
SM403	Выключен только на один программный цикл после RUN	Вкл. Выкл.	После установки режима RUN программа на один программный цикл устанавливается в состояние "выключено". Этот способ могут использовать только программы, выполняемые один раз за цикл.	С (обработка команды END)	M9039	
SM404	Включен только на один программный цикл после RUN	Вкл. Выкл.	После установки режима RUN программа на один программный цикл устанавливается в состояние "включено". Этот контакт могут использовать только программы, которые могут выполняться в замедленном режиме.	С (обработка команды END)	новый	● кроме Q00J, Q00 и Q01
SM405	Выключен только на один программный цикл после RUN	Вкл. Выкл.	После установки режима RUN программа на один программный цикл устанавливается в состояние "выключено". Этот контакт могут использовать только программы, которые могут выполняться в замедленном режиме.	С (обработка команды END)	новый	● кроме Q00J, Q00 и Q01
SM409	Такт 0.01 с		Повторяющееся изменение между включенным и выключенным состоянием с 10-миллисекундным интервалом. После отключения блока питания или сброса центрального процессора маркер автоматически переводится из выключенного состояния во включенное.	С (изменение состояния)	новый	Q, кроме Q00J, Q00 и Q01
SM410	Такт 0.1 с		Повторяющееся изменение между включенным и выключенным состоянием с указанным интервалом. Этот процесс продолжается и при состоянии STOP процессора. После отключения сетевого блока или сброса центрального процессора маркер автоматически переключается с выключенного состояния на включенное.	С (изменение состояния)	M9030	●
SM411	Такт 0.2 с				M9031	
SM412	Такт 1 с				M9032	
SM413	Такт 2 с				M9033	
SM414	Такт 2 x n с		Изменение между включенным и выключенным состоянием с интервалом в секундах, указанным в SD414.	С (изменение состояния)	M9034 изменился формат	
SM415	Такт 2 x n мс		Изменяется между включенным и выключенным состоянием с интервалом в миллисекундах, указанным в SD415.	С (изменение состояния)	новый	Q, кроме Q00J, Q00 и Q01

(3) Системные такты и счетчики

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM420	Такт № 0		Маркер непрерывно переключается между включенным и выключенным состоянием с заданным интервалом.  После отключения сетевого блока или сброса центрального процессора маркер автоматически переключается с выключенного состояния на включенное.  Длительность включенного и выключенного состояния устанавливается с помощью команды DUTY.	С (обработка команды END)	M9020	●
SM421	Такт № 1				M9021	
SM422	Такт № 2				M9022	
SM423	Такт № 3				M9023	
SM424	Такт № 4		M9024			
SM430	Такт № 5		Маркеры SM420...SM424 предназначены для программ с замедленным выполнением.	С (обработка команды END)	новый	● кроме Q00J, Q00 и Q01
SM431	Такт № 6					
SM432	Такт № 7					
SM433	Такт № 8					
SM434	Такт № 9					

(4) Информация цикла

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM510	Флаг выполнения программ с пониженной скоростью	выкл.: выполнение завершено или не выполняется вкл.: программа выполняется	Устанавливается при выполнении замедленных программ.	С (обработка команды END)	новый	● кроме Q00J, Q00 и Q01
SM551	Считывает интервал сервисного обслуживания модуля	выкл.: не выполняется вкл.: считывание	Если этот маркер изменяет состояние с выключенного на включенное, интервал сервисного обслуживания, указанный в регистре SD550, считывается в регистры SD550...SD551.	П	новый	



(5) Карты памяти

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" М9[ ][ ][ ]	Дейст. для:
SM600	Маркер применимости карты памяти "А"	выкл.: не применима вкл.: применима	Маркер установлен, если карта памяти "А" готова к применению пользователем.	С (инициализация)	новый	● кроме Q00J, Q00 и Q01
SM601	Маркер защиты карты памяти "А"	выкл.: не защищена вкл.: защищена	Маркер установлен, если установлен защитный выключатель карты памяти "А".	С (инициализация)	новый	
SM602	Маркер дисководов 1	выкл.: дисковод 1 не имеется вкл.: дисковод 1 имеется	Установлен, если дисковод 1 (карта 1, область RAM) имеется.	С (инициализация)	новый	
SM603	Маркер дисководов 2	выкл.: дисковод 2 не имеется вкл.: дисковод 2 имеется	Устанавливается, если дисковод 2 (карта 1, область ROM) имеется.	С (инициализация)	новый	
SM604	Маркер использования карты памяти "А"	выкл.: не используется вкл.: используется	Устанавливается, если карта памяти "А" используется.	С (инициализация)	новый	
SM605	Маркер блокировки карты памяти "А"	выкл.: вынимание/ вставление возможно вкл.: вынимание/ вставление запрещено	Устанавливается, если карту памяти "А" нельзя вынимать и вставлять.	П	новый	
SM620	Маркер применимости карты памяти "В"	выкл.: не применима вкл.: применима	Всегда включен	С (инициализация)	новый	Q
			Установлен, если карта памяти "В" готова к работе.	С (инициализация)	новый	Q2A(S1), Q3A, Q4A, Q4AR
SM621	Маркер защиты карты памяти "В"	выкл.: не защищена вкл.: защищена	Всегда включен.	С (инициализация)	новый	Q
			Устанавливается, если выключатель для защиты карты памяти "В" включен.	С (инициализация)	новый	Q2A(S1), Q3A, Q4A, Q4AR
SM622	Маркер дисководов 3	выкл.: дисковод 3 не имеется вкл.: дисковод 3 имеется	Всегда включен.	С (инициализация)	новый	Q
			Устанавливается, если дисковод 3 (карта 2, область RAM) имеется.	С (инициализация)	новый	Q2A(S1), Q3A, Q4A, Q4AR
SM623	Маркер дисководов 4	выкл.: дисковод 4 не имеется вкл.: дисковод 4 имеется	Всегда включен.	С (инициализация)	новый	Q
			Устанавливается, если дисковод 4 (карта 2, область ROM) имеется.	С (инициализация)	новый	Q2A(S1), Q3A, Q4A, Q4AR
SM624	Маркер использования карты памяти "В"	выкл.: не используется вкл.: используется	Устанавливается, если карта памяти "В" используется.	С (инициализация)	новый	
SM625	Маркер блокировки карты памяти "В"	выкл.: вынимание/ вставление возможно вкл.: вынимание/ вставление запрещено	Устанавливается, если карту памяти "В" нельзя вынимать и вставлять.	П	новый	

## (5) Карты памяти

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM640	Использование регистра файлов	выкл.: регистр файлов не используется вкл.: регистр файлов используется	Устанавливается, если регистр файлов используется.	С (изменение состояния)	новый	●
SM650	Использование комментария	выкл.: комментарий не используется вкл.: комментарий используется	Устанавливается, если файл комментариев используется.	С (изменение состояния)	новый	● кроме Q00J, Q00 и Q01
SM660	Процесс начальной загрузки	выкл.: выполнение из внутренней памяти вкл.: происходит процесс начальной загрузки	Установлен во время процесса начальной загрузки. Сбрасывается, если переключатель начальной загрузки не установлен.	С (изменение состояния)	новый	●
SM672	Маркер доступа к области регистров файлов на карте памяти "А"	выкл.: в пределах области доступа вкл.: за пределами области доступа	Маркер устанавливается при попытке доступа к областям карты памяти "А", находящимся вне области регистров файлов R. (Устанавливается при обработке команды END.)  Сбрасывается пользовательской программой.	С/П	новый	● кроме Q00J, Q00 и Q01
SM673	Маркер доступа к области регистров файлов на карте памяти "В"	выкл.: в пределах области доступа вкл.: за пределами области доступа	Маркер устанавливается при попытке доступа к областям карты памяти "В", находящимся вне области регистров файлов R. (Устанавливается при обработке команды END.)  Сбрасывается пользовательской программой.	С/П	новый	Q2A(S1), Q3A, Q4A, Q4AR

## (6) Маркеры диагностики, относящиеся к командам

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM700	Маркер переноса	выкл.: перенос выключен вкл.: перенос включен	Маркер переноса используется в прикладных командах.	С (выполнение команды)	M9012	●
SM701	Выбор выводимых знаков	выкл.: вывод 16 знаков вкл.: вывод до кода NUL	Если маркер установлен, выводятся 16 знаков в кодировке ASCII. Если маркер не установлен, вывод происходит до знака NUL (00 <sub>H</sub> ).	П	M9049	● кроме Q00J, Q00 и Q01
SM702	Метод поиска	выкл.: следующий поиск вкл.: 2-частный поиск	Предопределенный метод, используемый командами поиска. Для двухчастного поиска данные должны быть упорядочены.	П	новый	●
SM703	Команда сортировки	выкл.: по возрастанию вкл.: по убыванию	Для команд сортировки необходимо указывать направление сортировки.	П	новый	
SM704	Поблочное сравнение	выкл.: соответствия нет вкл.: полное соответствие	Устанавливается, если все условия данных совпадают с командой BKMP.	С (выполнение команды)	новый	

## (6) Маркеры диагностики, относящиеся к командам

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM707	Режим выполнения для команд, оперирующих числами с плавающей запятой.	выкл.: приоритетом является скорость вкл.: приоритетом является точность	Если SM707 не установлен, то команды, оперирующие числами с плавающей запятой, выполняются с высокой скоростью. При установленном SM707 команды, оперирующие числами с плавающей запятой, выполняются с высокой точностью.	П	новый	Q4AR
SM710	Приоритет проверки в команде CHK	выкл.: приоритет состояния контакта вкл.: приоритет проверочного контура	Если маркер не установлен, сохраняется имеющаяся настройка. Если маркер установлен, приоритет CHK обновляется.	С (выполнение команды)	новый	● кроме Q00J, Q00 и Q01
SM711	Состояние разделенной передачи	выкл.: во время иной обработки вкл.: во время разделенной обработки	Если работе с модулем AD57(S1), если передача ведется с разделенным экраном, этот маркер установлен в состояние "включено". По окончании передачи маркер сбрасывается.	С (выполнение команды)	M9065	QnA
SM712	Выбор обработки передачи	выкл.: пакетная передача вкл.: разделенная передача	При работе с модулем AD57(S1), если экран разделен для передачи, устанавливается этот маркер.	С (выполнение команды)	M9066	
SM714	Сигнал работы (сигнал BUSY) Область регистрации коммуникационного запроса	выкл.: коммуникационный запрос на модуль удаленного терминала возможен вкл.: коммуникационный запрос на модуль удаленного термин. не возможен	Выбор, должны ли выполняться запросы на коммуникацию с модулем удаленного терминала, соединенным с модулем AJ71PT32-S3 или процессором A2C.	С (выполнение команды)	M9081	
SM715	Маркер EI	выкл.: во время DI вкл.: во время EI	Устанавливается при выполнении команды EI.	С (выполнение команды)	новый	●
SM720	Комментарий считан	выкл.: считывание комментария не окончено вкл.: считывание комментария окончено	Этот маркер устанавливается после выполнения команды COMRD или PRC только на один цикл.	С (изменение состояния)	новый	Q, кроме Q00J, Q00 и Q01
SM721	Происходит обращение к файлу	выкл.: обращение к файлу не происходит вкл.: происходит обращение к файлу	Этот маркер устанавливается во время обращения к файлу с помощью команды S.FWRITE, S.FREAD, COMRD, PRC или LEDC.	С (изменение состояния)	новый	Q
SM722	Блокировка сообщений об ошибке для команд BIN и DBIN	выкл.: сообщения об ошибке деблокированы вкл.: сообщения об ошибке заблокированы	Чтобы заблокировать сообщения об ошибке выполнения команд BIN и DBIN, необходимо установить этот маркер.	П	новый	
SM730	Сигнал BUSY для области коммуникационного запроса CC-Link	выкл.: запрос на обмен данными со спец. модулем деблокирован вкл.: запрос на обмен данными заблокирован	Этот маркер используется для деблокировки и блокировки сигнала на запрос коммуникации со специальным модулем, подключенным к A(1S)J61QBT11.	С (выполнение команды)	новый	QnA

## (6) Маркеры диагностики, относящиеся к командам

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" М9[ ][ ][ ]	Дейст. для:
SM736	Маркер выполнения команды PKEY	выкл.: команда не выполняется вкл.: команда выполняется	Устанавливается при выполнении команды PKEY. Сбрасывается при вводе кода CR или если длина строковой величины достигла 32 знаков.	С (выполнение команды)	новый	● кроме Q00, Q00 и Q01
SM737	Маркер приема ввода с клавиатуры, сделанного с помощью команды PKEY	выкл.: прием ввода с клавиатуры возможен вкл.: прием ввода с клавиатуры не возможен	Устанавливается, если был сделан ввод с клавиатуры. Сбрасывается после того, как центральный процессор сохранил ввод с клавиатуры.	С (выполнение команды)	новый	
SM738	Маркер приема команды MSG	выкл.: команда не выполняется вкл.: команда выполняется	Устанавливается при выполнении команды MSG.	С (выполнение команды)	новый	
SM774	ПИД-регулирование с плавной характеристикой регулирования	выкл.: принудительное соответствие вкл.: соответствие не устанавливается принудительно	Выбор, устанавливается ли принудительно соответствие между фактическим и заданным значением в ручном режиме.	П	новый	
SM775	Выбор обработки обновления связи во время выполнения команды COM	выкл.: обновление связи выполняется вкл.: обновление связи не выполняется	Если маркер не установлен, происходит обновление сетевых и интерфейсных данных (обновление канала связи) и обработка всех данных (обработка команды END). Если маркер установлен, происходит только обработка всех данных (обработка команды END).	П	новый	●
SM776	Деблокировка или блокировка локальных устройств при команде CALL	выкл.: локальные устройства заблокированы вкл.: локальные устройства деблокированы	Выбор, деблокируются ли локальные устройства в программе, вызываемой с помощью команды CALL.	П (изменение состояния)	новый	● кроме Q00, Q00 и Q01
SM777	Деблокировка или блокировка локальных устройств во время программы прерывания	выкл.: локальные устройства заблокированы вкл.: локальные устройства деблокированы	Выбор, деблокируются ли локальные устройства во время выполнения программы прерывания.	П (изменение состояния)	новый	
SM780	Выполнимость команд, относящихся к CC-Link	выкл.: выполнение команд, относящихся к CC-Link, возможно вкл.: выполнение команд, относящ. к CC-Link, не возможно	Этот маркер устанавливается, если достигнуто максимальное количество относящихся к CC-Link команд, которое может выполняться одновременно (32 команды). Этот маркер сбрасывается, если число команд снова снизилось ниже 32.	С (изменение состояния)	новый	QnA

## (7) Устранение ошибок

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM800	Подготовка трассировки	выкл.: не подготовлена вкл.: подготовка завершена	Устанавливается, если подготовка трассировки завершена.	С (изменение состояния)	новый	Q, кроме Q00J, Q00 и Q01
	Подготовка выборочной трассировки			С (изменение состояния)		QnA
SM801	Трассировка запущена	выкл.: останов трассировки/ выборочной трассировки	Этот маркер устанавливается для запуска трассировки или выборочной трассировки и сбрасывается для ее останова. (в сочетании со всеми сброшенными маркерами диагностики)	П	M9047	Q, кроме Q00J, Q00 и Q01
	Запуск выборочной трассировки	вкл.: запуск трассировки/ выборочной трассировки		П		QnA
SM802	Процесс выполнения трассировки	выкл.: останов трассировки/ выборочной трассировки	Устанавливается во время выполнения трассировки/ выборочной трассировки.	С (изменение состояния)	M9046	Q, кроме Q00J, Q00 и Q01
	Процесс выполнения выборочной трассировки	вкл.: запуск трассировки/ выборочной трассировки		С (изменение состояния)		QnA
SM803	Триггер трассировки	выкл. → вкл.: запуск	Функция этого маркера аналогична функции команды TRACE. Триггер выборочной трассировки устанавливается при изменении состояния маркера с "Выкл." на "Вкл."	П	M9044	Q, кроме Q00J, Q00 и Q01
	Триггер выборочной трассировки		Функция этого маркера аналогична функции команды STRA. Триггер выборочной трассировки устанавливается при изменении состояния маркера с "Выкл." на "Вкл."	П		QnA
SM804	После триггера трассировки	выкл.: не после триггера вкл.: после триггера	Этот маркер устанавливается после триггера трассировки.	С (изменение состояния)	новый	Q, кроме Q00J, Q00 и Q01
	После триггера выборочной трассировки		Этот маркер устанавливается после триггера выборочной трассировки.	С (изменение состояния)		QnA
SM805	Трассировка окончена		Этот маркер устанавливается по окончании трассировки.	С (изменение состояния)	M9043	Q, кроме Q00J, Q00 и Q01
	Выборочная трассировка окончена	выкл.: происходит трассировка/ выборочная трассировка вкл.: трассировка/ выборочная трассировка окончена	Этот маркер устанавливается по окончании выборочной трассировки.	С (изменение состояния)		QnA

## (7) Устранение ошибок

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM806	Подготовка фиксации состояния	выкл.: не подготовлена вкл.: подготовлена	Устанавливается по окончании подготовки.	С (изменение состояния)	новый	QnA
SM807	Команда фиксации состояния	выкл. → вкл.: фиксация	Выполнение команды фиксации состояния	П	новый	
SM808	Фиксация состояния завершена	выкл.: фиксация сост. не завершена вкл.: фиксация состояния завершена	Устанавливается по окончании фиксации состояния.	С (изменение состояния)	M9055	
SM809	Фиксация состояния свободна	выкл. → вкл.: свободна	Возможна очередная фиксация состояния.	П	новый	
SM810	Подготовка контроля программы	выкл.: не подготовлен вкл.: подготовка завершена	Устанавливается по окончании подготовки к контролю программы.	С (изменение состояния)	новый	
SM811	Запуск контроля программы	выкл.: останов контроля программы вкл.: запуск контроля программы	Этот маркер устанавливается для запуска контроля программы и сбрасывается для его останова. (в сочетании со всеми сброшенными маркерами диагностики)	П	новый	QnA
SM812	Происходит контроль программы	выкл.: останов контроля программы вкл.: запуск контроля программы	Установлен при выполнении контроля программы.	С (изменение состояния)	новый	
SM813	Триггер контроля программы	выкл. → вкл.: запуск	Функция этого маркера аналогична функции команды PTRА. Триггер контроля программы устанавливается при переключении маркера из выключенного во включенное состояние.	П	новый	
SM814	После триггера контроля программы	выкл.: не после триггера вкл.: после триггера	Этот маркер устанавливается после триггера контроля программы.	С (изменение состояния)	новый	
SM815	Контроль программы завершен	выкл.: происходит контроль прогр. вкл.: контроль прогр. завершен	Этот маркер устанавливается по окончании контроля программы.	С (изменение состояния)	новый	
SM820	Подготовка контроля шага	выкл.: не подготовлен вкл.: подготовка завершена	Устанавливается по окончании подготовки к контролю шага.	С (изменение состояния)	новый	

## (7) Устранение ошибок

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM821	Запуск контроля шага	выкл.: останов контроля шага вкл.: запуск контроля шага	Этот маркер устанавливается для запуска контроля шага и сбрасывается для останова. (в сочетании со всеми сброшенными маркерами диагностики)	П	M9182 изменился формат	● кроме Q00J, Q00 и Q01
SM822	Происходит контроль шага	выкл.: останов контроля шага вкл.: запуск контроля шага	Устанавливается при выполнении контроля шага. Сбрасывается при завершении или приостановке контроля шага.	С (изменение состояния)	M9181	
SM823	После триггера контроля шага одного блока	выкл.: не после триггера вкл.: после первого триггера	Маркер устанавливается после срабатывания триггера первого блока, в котором был выполнен контроль шага. Маркер сбрасывается, если начался контроль шага.	С (изменение состояния)	новый	
SM824	После триггера контроля шага всех блоков	выкл.: не после триггера вкл.: после всех триггеров	Маркер устанавливается после срабатывания триггеров всех блоков, в которых был выполнен контроль шага. Маркер сбрасывается, если начался контроль шага.	С (изменение состояния)	новый	
SM825	Контроль шага завершен	выкл.: происходит контроль шага вкл.: контроль шага завершен	Этот маркер устанавливается по окончании контроля шага. Этот маркер сбрасывается при начале контроля шага.	С (изменение состояния)	M9180	
SM826	Ошибка при трассировке	выкл.: норм. состояние вкл.: ошибка	Этот маркер устанавливается, если во время трассировки возникла ошибка.	С (изменение состояния)	новый	Q, кроме Q00J, Q00 и Q01
	Ошибка при выборочной трассировке		Маркер устанавливается, если во время выборочной трассировки возникла ошибка.	С (изменение состояния)		
SM827	Ошибка при фиксации состояния	выкл.: норм. состояние вкл.: ошибка	Маркер устанавливается, если во время фиксации состояния возникла ошибка.	С (изменение состояния)	новый	QnA
SM828	Ошибка при контроле программы	выкл.: норм. состояние вкл.: ошибка	Маркер устанавливается, если во время контроля программы возникла ошибка.	С (изменение состояния)	новый	

## (8) Область фиксации

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Серия "А" M9[ ][ ][ ]	Дейст. для:
SM900	Обработка файла во время падения напряжения	выкл.: файл не обрабатывался во время падения напряжения вкл.: файл обрабатывался во время падения напряжения	Этот маркер устанавливается, если во время обработки файла произошло падение напряжения.	С/П (изменение состояния)	новый	QnA
SM910	Флаг регистрации PKEY	выкл.: ввод с клавиатуры не зарегистрирован вкл.: ввод с клавиатуры зарегистрирован	Этот маркер устанавливается при регистрации ввода с клавиатуры. Маркер сбрасывается, если ввод с клавиатуры не был зарегистрирован.	С (выполнение команды)	новый	● кроме Q00J, Q00 и Q01

(9) Соответствия между специальными маркерами (серии "А") и маркерами диагностики (серии QnA / "System Q")

При переходе с контроллеров MELSEC серии "А" на контроллеры MELSEC серии "Q" или "System Q" специальные маркеры M9000...M9255 (MELSEC серии "А") соответствуют маркерам диагностики SM1000...SM1255 (MELSEC серии "Q").

Все эти маркеры диагностики устанавливает система – их изменение с помощью пользовательской программы не возможно. Пользователи, желающие устанавливать или сбрасывать эти маркеры, должны изменить свои программы таким образом, чтобы использовались только настоящие маркеры диагностики серии QnA. Исключением являются специальные маркеры M9084 и M9200...M9255. Если перед переходом на контроллеры MELSEC серии "Q"/"System Q" была возможна установка и сброс этих маркеров, то после перехода можно устанавливать и сбрасывать и соответствующие маркеры диагностики SM1084 и SM1200...SM1255.

Подробную информацию о специальных маркерах серии "А" можно найти в руководствах по центральным процессорам и сетям "MELSECNET" и "MELSECNET/B".

**ПРИМЕЧАНИЕ** При использовании конвертированных специальных маркеров время обработки в центральном процессоре Q может увеличиться. Если конвертированные специальные маркеры не используются, то в среде программирования (в параметрах контроллера, закладка "Система контроллера") следует отменить выбор опции "Контроллер А: использование специальных маркеров и регистров SM/SD 1000".

Если в таблице указан эквивалентный маркер диагностики для центральных процессоров "System Q" или QnA, то следует изменить программу и использовать этот маркер. Если эквивалентный маркер диагностики для "System Q"/QnA не указан, можно использовать конвертированный маркер.

#### Перечень специальных маркеров и маркеров диагностики

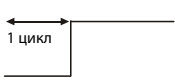
Специальный маркер процессора серии "А"	Конвертированный маркер диагностики	Эквивалентный маркер диагностики "System Q"/QnA	Название	Значение	Дейст. для:
M9000	SM1000	—	Неисправен предохранитель	выкл.: нормальное состояние вкл.: ошибка	Q и QnA
M9002	SM1002	—	Ошибка сверки модуля ввода-вывода	выкл.: нормальное состояние вкл.: ошибка	
M9004	SM1004	—	Неисправность в мастер-модуле MELSECNET MINI	выкл.: нормальное состояние вкл.: ошибка	QnA
M9005	SM1005	—	Падение сетевого напряжения	выкл.: нормальное состояние вкл.: падение напряжения	Q и QnA
M9006	SM1006	—	Низкое напряжение батареи	выкл.: нормальное состояние вкл.: падение напряжения	
M9007	SM1007	—	Низкое напряжение батареи (фиксируемый маркер)	выкл.: нормальное состояние вкл.: падение напряжения	
M9008	SM1008	SM1	Распознавание ошибки после самодиагностики	выкл.: нормальное состояние вкл.: ошибка	
M9009	SM1009	SM62	Маркер наличия маркера ошибки	выкл.: не обнаружен вкл.: обнаружен	
M9011	SM1011	SM56	Распознавание ошибки в процессе выполнения программы	выкл.: нормальное состояние вкл.: ошибка	
M9012	SM1012	SM700	Флаг переноса (маркер переноса)	выкл.: перенос выключен вкл.: перенос включен	
M9016	SM1016	не действует в процессорах Q и QnA	Маркер стирания сохраненных данных операндов	выкл.: не выполняется вкл.: процесс стирания	



Перечень специальных маркеров и маркеров диагностики

Специальный маркер процессора серии "А"	Конвертированный маркер диагностики	Эквивалентный маркер диагностики "System Q"/QnA	Название	Значение	Дейст. для:
M9017	SM1017	в процессорах "System Q" и QnA не задействован	Маркер стирания сохраненных данных операндов	выкл.: не выполняется вкл.: стирание	Q и QnA
M9020	SM1020	—	Такт № 0		
M9021	SM1021	—	Такт № 1		
M9022	SM1022	—	Такт № 2		
M9023	SM1023	—	Такт № 3		
M9024	SM1024	—	Такт № 4		
M9025	SM1025	—	Запрос на установку данных часов	выкл.: не обрабатывается вкл.: запрос	
M9026	SM1026	—	Ошибка данных часов	выкл.: нормальное состояние вкл.: ошибка	
M9027	SM1027	—	Индикация данных часов	выкл.: не обрабатывается вкл.: запрос	
M9028	SM1028	—	Запрос на считывание данных часов	выкл.: нормальное состояние вкл.: ошибка	
M9029	SM1029	в процессорах "System Q" и QnA не задействован	Пакетная обработка данных коммуникационного запроса	выкл.: пакетная обработка не выполняется вкл.: пакетная обработка выполняется	
M9030	SM1030	—	Датчик тактовых импульсов 0.1 секунды		
M9031	SM1031	—	Датчик тактовых импульсов 0.2 секунды		
M9032	SM1032	—	Датчик тактовых импульсов 1 секунда		
M9033	SM1033	—	Датчик тактовых импульсов 2 секунды		
M9034	SM1034	—	Датчик тактовых импульсов 1 минута		
M9036	SM1036	—	Постоянно включен	вкл. ————— выкл. —————	
M9037	SM1037	—	Постоянно выключен	вкл. ————— выкл. —————	
M9038	SM1038	—	Включен на 1 цикл только после RUN	вкл. ————— выкл. ————— 	

## Перечень специальных маркеров и маркеров диагностики

Специальный маркер процессора серии "А"	Конвертированный маркер диагностики	Эквивалентный маркер диагностики "System Q"/QnA	Название	Значение	Дейст. для:
M9039	SM1039	—	Выключен только на 1 цикл после RUN	вкл.  выкл.	Q и QnA
M9040	SM1040	SM206	Условие паузы	выкл.: режим PAUSE не возможен вкл.: режим PAUSE возможен	
M9041	SM1041	SM204	Маркер состояния PAUSE	выкл.: PAUSE не имеет места вкл.: во время состояния PAUSE	
M9042	SM1042	SM203	Маркер состояния STOP	выкл.: STOP не имеет места вкл.: во время состояния STOP	
M9043	SM1043	SM805	Выборочная трассировка окончена	выкл.: во время выборочной трассировки вкл.: по окончании выборочной трассировки	
M9044	SM1044	SM803	Выборочная трассировка	0 → 1 аналогично выполнению команды STRA 1 → 0 аналогично выполнению команды STRAR	
M9045	SM1045	в процессорах "System Q" и QnA не задействован	Сброс контрольного таймера	выкл.: сброса нет вкл.: контрольный таймер сбрасывается	
M9046	SM1046	SM802	Выборочная трассировка	выкл.: контроль не активен вкл.: контроль активен	
M9047	SM1047	SM801	Подготовка выборочной трассировки	выкл.: останов выборочной трассировки вкл.: запуск выборочной трассировки	
M9049	SM1049	SM701	Количество выводимых знаков	выкл.: вывод до кода NUL вкл.: вывод 16 знаков	
M9051	SM1051	в процессорах "System Q" и QnA не задействован	Подавление команды CHG	выкл.: выполнение возможно вкл.: выполнение не возможно	
M9052	SM1052	в процессорах "System Q" и QnA не задействован	Переключение команды SEG	выкл.: 7-сегментный индикатор вкл.: частичное обновление ввода-вывода	
M9054	SM1054	SM205	Маркер режима "STEP RUN"	выкл.: иной режим вкл.: STEP RUN	
M9055	SM1055	SM808	Маркер фиксации состояния	выкл.: не окончена вкл.: окончена	
M9055	SM1055	SM808	Маркер фиксации состояния	выкл.: не окончена вкл.: окончена	
M9056	SM1056	в процессорах "System Q" и QnA не задействован	Запрос P, I для основной программы	выкл.: без запроса вкл.: запрос P, I	Q и QnA
M9057	SM1057		Запрос P, I для подпрограммы	выкл.: без запроса вкл.: запрос P, I	
M9058	SM1058		Основная программа, установка P, I завершена.	Кратковременно включается после завершения установки P, I.	
M9059	SM1059		Подпрограмма, установка P, I завершена.	Кратковременно включается после завершения установки P, I.	
M9060	SM1060		Запрос P, I для подпрограммы 2	выкл.: без запроса вкл.: запрос P, I	
M9061	SM1061		Запрос P, I для подпрограммы 3	выкл.: без запроса вкл.: запрос P, I	

## Перечень специальных маркеров и маркеров диагностики

Специальный маркер процессора серии "А"	Конвертированный маркер диагностики	Эквивалентный маркер диагностики "System Q"/QnA	Название	Значение	Дейст. для:
M9065	SM1065	SM711	Маркер пошаговой передачи	выкл.: иная обработка вкл.: пошаговая передача	QnA
M9066	SM1066	SM712	Переключение обработки передачи	выкл.: пакетная передача вкл.: пошаговая передача	
M9070	SM1070	в процессорах "System Q" и QnA не задействован	A8UPU/A8PUJ Требуемое время поиска	выкл.: время считывания не сокращено вкл.: время считывания сокращено	Q и QnA
M9081	SM1081	SM714	Коммуникационный запрос на удаленный специальный модуль	выкл.: запрос возможен вкл.: запрос не возможен	QnA
M9084	SM1084	в процессорах "System Q" и QnA не задействован	Контроль на ошибки	выкл.: контроль на ошибки выполняется вкл.: без контроля на ошибки	Q и QnA
M9091	SM1091		Маркер ошибки команды	выкл.: нормальное состояние вкл.: ошибка	
M9094	SM1094	SM251	Маркер изменения модулей ввода-вывода	выкл.: изменение имеется вкл.: изменения нет	QnA
M9100	SM1100	SM320	Наличие/отсутствие программы на языке SFC	выкл.: программы на языке SFC не используются вкл.: программы на языке SFC используются	Q и QnA
M9101	SM1101	SM321	Запуск/останов программы на языке SFC	выкл.: останов программы на языке SFC вкл.: запуск программ на языке SFC	
M9102	SM1102	SM322	Вид запуска программы на языке SFC	выкл.: первоначальный запуск: вкл.: возобновление	
M9103	SM1103	SM323	Наличие/отсутствие непрерывных переходов	выкл.: переход не действует вкл.: переход действует	
M9104	SM1104	SM324	Флаг индикации непрерывного перехода	выкл.: при выполненном переходе вкл.: переход не происходит	
M9108	SM1108	SM90	Запуск контрольного таймера пошагового перехода (эквивалентно D9108)	выкл.: контрольный таймер сброшен вкл.: запуск сброса контрольного таймера	
M9109	SM1109	SM91	Запуск контрольного таймера пошагового перехода (эквивалентно D9109)		
M9110	SM1110	SM92	Запуск контрольного таймера пошагового перехода (эквивалентно D9110)		
M9111	SM1111	SM93	Запуск контрольного таймера пошагового перехода (эквивалентно D9111)		
M9112	SM1112	SM94	Запуск контрольного таймера пошагового перехода (эквивалентно D9112)		
M9113	SM1113	SM95	Запуск контрольного таймера пошагового перехода (эквивалентно D9113)		
M9114	SM1114	SM96	Запуск контрольного таймера пошагового перехода (эквивалентно D9114)		

## Перечень специальных маркеров и маркеров диагностики

Специальный маркер процессора серии "А"	Конвертированный маркер диагностики	Эквивалентный маркер диагностики "System Q"/QnA	Название	Значение	Дейст. для:
M9180	SM1180	SM825	Флаг завершения семплирования активного шага	выкл.: семплирование запускается вкл.: семплирование завершено	Q и QnA
M9181	SM1181	SM822	Флаг выполнения семплирования активного шага	выкл.: семплирование не выполняется вкл.: семплирование выполняется в данный момент	
M9182	SM1182	SM821	Деблокировка семплирования активного шага	выкл.: семплирование не возможно или приостановлено вкл.: семплирование возможно	
M9196	SM1196	SM325	Вывод рабочего шага после останова блока	выкл.: выходы выкл. вкл.: выходы вкл.	
M9197 M9198	SM1197 SM1198	в процессорах "System Q" и QnA не задействован	Переключение между индикацией неисправности предохранителя и индикацией ошибки сверки модуля ввода-вывода	Индикация изменяется в зависимости от сочетания состояния маркеров M9197 и M9198	
M9199	SM1199		Онлайн-регистрация данных фиксации состояния выборочной трассировки	выкл.: регистрация данных не происходит вкл.: регистрация данных происходит	
M9200	SM1200	—	Прием команды LRDP	выкл.: не принята вкл.: принята	QnA
M9201	SM1201	—	Обработка команды LRDP	выкл.: не завершена вкл.: завершена	
M9202	SM1202	—	Прием команды LWTP	выкл.: не принята вкл.: принята	
M9203	SM1203	—	Обработка команды LWTP	выкл.: не завершена вкл.: завершена	
M9204	SM1204	—	Обработка команды LRDP	выкл.: не завершена вкл.: завершена	
M9205	SM1205	—	Обработка команды LWTP	выкл.: не завершена вкл.: завершена	
M9206	SM1206	—	Ошибка в параметрах связи хост-станции	выкл.: нормальное состояние вкл.: ошибка	
M9207	SM1207	—	Соответствие параметров связи нескольких мастер-станций	выкл.: нормальное состояние вкл.: соответствия нет	
M9208	SM1208	—	Диапазон передачи В и W для мастер-станции на нижнем уровне	выкл.: на 2-й и 3-й ярус вкл.: только на 2-й ярус	
M9208	SM1208	—	Диапазон передачи В и W для мастер-станции на нижнем уровне	выкл.: на 2-й и 3-й ярус вкл.: только на 2-й ярус	
M9209	SM1209	—	Проверка параметров связи (только для мастер-станций на нижнем уровне)	выкл.: проверка вкл.: без проверки	
M9210	SM1210	—	Ошибка карты связи в локальной станции	выкл.: нормальное состояние вкл.: ошибка	
M9211	SM1211	—	Ошибка карты связи в мастер-станции	выкл.: нормальное состояние вкл.: ошибка	

## Перечень специальных маркеров и маркеров диагностики

Специальный маркер процессора серии "А"	Конvertированный маркер диагностики	Эквивалентный маркер диагностики "System Q"/QnA	Название	Значение	Дейст. для:
M9224	SM1224	—	Состояние связи	выкл.: онлайн вкл.: офлайн	QnA
M9225	SM1225	—	Ошибка в прямой петле	выкл.: нормальное состояние вкл.: ошибка	
M9226	SM1226	—	Ошибка в обратной петле	выкл.: нормальное состояние вкл.: ошибка	
M9227	SM1227	—	Состояние тестирования петли	выкл.: тестирование не происходит вкл.: тестирование прямой или обратной петли	
M9232	SM1232	—	Рабочее состояние локальной станции	выкл.: RUN или STEP RUN вкл.: STOP или PAUSE	
M9233	SM1233	—	Распознавание ошибки для локальной станции	выкл.: нормальное состояние вкл.: ошибка	
M9235	SM1235	—	Ошибка параметра в локальной или удаленной станции ввода-вывода	выкл.: нормальное состояние вкл.: ошибка	
M9236	SM1236	—	Состояние инициализации локальной или удаленной станции ввода-вывода	выкл.: передача не происходит вкл.: передача данных	
M9237	SM1237	—	Ошибка в локальной или удаленной станции ввода-вывода	выкл.: нормальное состояние вкл.: ошибка	
M9238	SM1238	—	Ошибка в петле локальной или удаленной станции ввода-вывода	выкл.: нормальное состояние вкл.: ошибка	
M9240	SM1240	—	Состояние связи	выкл.: онлайн вкл.: офлайн	
M9241	SM1241	—	Ошибка в прямой петле	выкл.: нормальное состояние вкл.: ошибка	
M9242	SM1242	—	Ошибка в обратной петле	выкл.: нормальное состояние вкл.: ошибка	
M9243	SM1243	—	Передача по обратной петле	выкл.: не выполняется вкл.: выполняется	
M9246	SM1246	—	Состояние приема данных	выкл.: данные приняты вкл.: данные не приняты	
M9247	SM1247	—	Состояние приема данных	выкл.: данные приняты вкл.: данные не приняты	
M9250	SM1250	—	Состояние приема параметров	выкл.: параметры приняты вкл.: параметры не приняты	
M9251	SM1251	—	Обрыв передачи	выкл.: нормальное состояние вкл.: обрыв	
M9252	SM1252	—	Состояние тестирования петли	выкл.: тестирование не происходит вкл.: тестирование прямой или обратной петли	
M9253	SM1253	—	Рабочее состояние мастер-станции	выкл.: RUN или STEP RUN вкл.: STOP или PAUSE	
M9254	SM1254	—	Рабочее состояние другой локальной станции	выкл.: RUN или STEP RUN вкл.: STOP или PAUSE	
M9255	SM1255	—	Распознавание ошибки для других локальных станций	выкл.: нормальное состояние вкл.: ошибка	

## А.4.2 Перечень специальных маркеров (серии "А")

Специальные маркеры – это внутренние маркеры, которые можно применять для множества разнообразных целей, например, в качестве индикаторов ошибок, для специальных функций и т. п. В следующей таблице дан обзор всех специальных маркеров контроллеров MELSEC серии "А" и их назначения.

В принципе, можно различать два вида специальных маркеров:

- специальные маркеры, автоматически устанавливаемые центральным процессором – пользователь может их только выключать (сбрасывать);
- специальные маркеры, которые могут устанавливаться и сбрасываться при определенных условиях в зависимости от их функции.

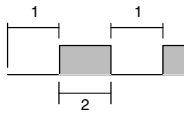
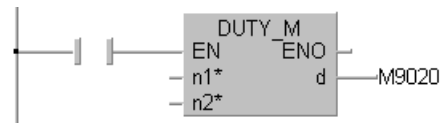
**ПРИМЕЧАНИЯ** *Область применения специальных маркеров в основной программе следует контролировать.*

*Маркеры, которые в столбце "Адресов" помечены символами ①, ② или ③, не могут произвольно устанавливаться или сбрасываться. Соответствующие примечания для этих маркеров находятся за таблицей на стр. 40.*


*В какой мере эти специальные маркеры могут применяться для определенных центральных процессоров, указано символами, разъясненными в следующей таблице.*

Центральный процессор		Значение
		неограниченное применение для центральных процессоров всех типов
○	(AnA)	кроме указанного процессора (процессоров)
●	A2C	только для указанного процессора (процессоров)

Перечень специальных маркеров

Адрес	Значение	Состояние	Описание	Процессор	
● M9000	Неисправен предохранитель	выкл.: нормальное состояние вкл.: неисправен	Этот маркер устанавливается, как только в одном или нескольких модулях обнаруживается неисправность предохранителя. Маркер остается установленным и после возврата в нормальное состояние.	○	(кроме A2C)
● M9002	Ошибка сверки модулей ввода-вывода	выкл.: нормальное состояние вкл.: ошибка	Текущее состояние сети A2C отличается от состояния после включения напряжения питания. Маркер остается установленным и после исчезновения ошибки. Сброс маркера возможен только после сброса регистров с D9116 по D9123.	○	(кроме A2C)
M9004	Ошибка в мастер-модуле des MELSECNET/MINI	выкл.: нормальное состояние вкл.: ошибка	Этот маркер устанавливается при наличии ошибки в одном из главных модулей типа AJ71PT32(S3) в MELSECNET/MINI. Маркер остается установленным и после исчезновения ошибки.	●	только для AnA, AnAS, AnU
● M9005	Падение сетевого напряжения	выкл.: нормальное состояние вкл.: падение напряжения	Этот маркер устанавливается при падении напряжения на время не более 20 мс. Сброс возможен после выключения и повторного включения электропитания.		
M9006	Низкое напряжение батареи	выкл.: нормальное состояние вкл.: напряжение упало	Напряжение батареи буферного питания снизилось ниже минимального предела. После замены батареи маркер сбрасывается.		
● M9007	Низкое напряжение батареи (фиксируемый маркер)	выкл.: нормальное состояние вкл.: напряжение упало	Напряжение батареи буферного питания снизилось ниже минимального предела. После замены батареи маркер остается установленным.		
● M9008	Распознавание ошибки после самодиагностики	выкл.: нормальное состояние вкл.: ошибка	Центральный процессор распознал ошибку с помощью функции самодиагностики.		
M9009	Маркер наличия маркера ошибки	выкл.: не обнаружен вкл.: обнаружен	Этот маркер устанавливается после выполнения команды OUTF или SET F. Сброс маркера происходит после установки содержимого регистра D9124 на „0“.		
M9010	Распознавание ошибки в процессе выполнения программы	выкл.: нормальное состояние вкл.: ошибка	При работе программы возникла ошибка при выполнении прикладной команды. После устранения ошибки маркер сбрасывается.	○	(кроме A3H, A3M, AnA, AnAS, AnU)
● M9011	Распознавание ошибки в процессе выполнения программы	выкл.: нормальное состояние вкл.: ошибка	При работе программы возникла ошибка при выполнении прикладной команды. После устранения ошибки маркер остается установленным.		
M9012	Флаг переноса (перенос)	выкл.: перенос выключен вкл.: перенос включен	Флаг переноса, используемый в прикладных командах.		
M9016	Маркер стирания сохраненных данных операндов	выкл.: не происходит вкл.: стирание	Если M9016 = „1“, все данные операндов (за исключением специальных маркеров и специальных регистров) в режиме Remote-RUN контроллера стираются.		
M9017	Маркер стирания сохраненных данных операндов	выкл.: не происходит вкл.: стирание	Если M9017 = „1“, то все данные операндов, которые не были промежуточно сохранены, (за исключением специальных маркеров и специальных регистров) стираются в режиме Remote-RUN контроллера.		
M9020	Такт № 0	 <p>1 цикл n2 2 цикл n1</p>	<p>Включение и выключение маркеров происходит с цикличностью, установленной с помощью команды DUTY (например, 2 цикла включены, 3 цикла выключены, 2 включены, 3 выключены и т. п.)</p> <p>После включения или сброса маркер начинает переключения с состояния „0“.</p> 		
M9021	Такт № 1				
M9022	Такт № 2				
M9023	Такт № 3				
M9024	Такт № 4				
● M9025	Запрос на установку данных часов	выкл.: не обрабатывается вкл.: запрос	При установленном маркере после выполнение команды END данных часов считываются из регистров данных с D9025 по D9028 и передаются в часы.	●	только для AnN, AnS

## Перечень специальных маркеров

Адрес	Значение	Состояние	Описание	Процессор
M9026	Ошибка в данных часов	выкл.: нормальное состояние вкл.: ошибка	Этот маркер устанавливается при ошибочных данных часов (регистры с D9025 по D9028).	● только для AnN, AnS
M9027	Индикация данных часов	выкл.: не обрабатывается вкл.: индикация	Данные часов считываются из регистров данных с D9025 по D9028 и выводятся в формате "месяц, день, час, минута и секунда" на светодиодный дисплей центрального процессора.	● только для A3N, A3A
 M9028	Запрос на считывание данных часов	выкл.: не обрабатывается вкл.: запрос	В регистры данных с D9025 по D9028 считываются данные часов в двоично-десятичном формате.	● только для AnN, AnS
M9030	Датчик тактовых импульсов 0.1 секунд	выкл.: 0.05 с вкл.: 0.05 с	Датчики тактовых импульсов для тактов от 0,1 до 60 секунд. Включение и отключение происходит не при каждом прогоне программы, а с выбранной циклическостью, даже в пределах одного прогона. Датчики тактовых импульсов запускаются непосредственно после включения электропитания или сброса.	
M9031	Датчик тактовых импульсов 0.2 секунды	выкл.: 0.1 с вкл.: 0.1 с		
M9032	Датчик тактовых импульсов 1 секунда	выкл.: 0.5 с вкл.: 0.5 с		
M9033	Датчик тактовых импульсов 2 секунды	выкл.: 1 с вкл.: 1 с		
M9034	Датчик тактовых импульсов 1 минута	выкл.: 30 с вкл.: 30 с		
M9036	Фиктивный контакт	Постоянно включен	Маркеры с M9036 по M9039 являются так называемыми фиктивными контактами для инициализирующей части и для прикладных команд в основной программе. Маркеры M9036 и M9037 устанавливаются независимо от положения выключателя с ключом на модуле центрального процессора. Маркеры M9038 и M9039 устанавливаются и сбрасываются в зависимости от состояния (RUN или STOP) центрального процессора. M9038 включается на один программный цикл после переключения в состояние RUN из состояния STOP. Маркер M9039 ведет себя наоборот.	
M9037	Фиктивный контакт	Постоянно выключен		
M9038	Маркер режима RUN (закрывающий контакт)	вкл.: на один цикл после RUN выкл.: после 1 цикла		
M9039	Маркер режима RUN (размыкающий контакт)	вкл.: после 1 цикла выкл.: на один цикл после RUN		
M9040	Условие паузы	выкл.: режим PAUSE не возможен вкл.: режим PAUSE возможен	Переключение центрального процессора в состояние PAUSE (путем переключения выключателя с ключом в положение "PAUSE" или подачи внешнего сигнала PAUSE) возможно только при установленном маркере M9040. Если центральный процессор находится в состоянии PAUSE, устанавливается маркер M9041.	
M9041	Маркер состояния "PAUSE"	выкл.: не в состоянии PAUSE вкл.: во время состояния PAUSE		
M9042	Маркер состояния STOP	выкл.: не в состоянии STOP вкл.: в состоянии STOP	Этот маркер устанавливается, как только выключатель с ключом на модуле центрального процессора переводится в положение "STOP".	
M9043	Выборочная трассировка окончена	выкл.: во время выборочн. трассировки вкл.: по окончании выборочн. трассировки	Маркер устанавливается при выполнении команды STRA по окончании выборочной трассировки. При выполнении команды STRAR маркер сбрасывается.	○ (кроме A1, A1N)
M9044	Выборочная трассировка	0→1: соответствует команде STRA 1→0: соответствует команде STRAR	Функция этого маркера аналогична командам STRA и STRAR (маркер M9044 можно установить с помощью программатора). Установка маркера соответствует выполнению команды STRA, а сброс – выполнению команды STRAR. Состояние обработки выборочной трассировки здесь зависит от значения регистра D9044 („0" для пошагового опроса с приращением 10 мс).	● только для A2C
M9046	Выборочная трассировка	выкл.: нормальное состояние вкл.: во время выборочной трассировки	Маркер выборочной трассировки; устанавливается при ее выполнении.	○ (кроме A1, A1N)
M9047	Подготовка выборочной трассировки	выкл.: останов выборочной трассировки вкл.: запуск выборочной трассировки	Этот маркер устанавливается для запуска выборочной трассировки и сбрасывается для ее останова.	○ (кроме A1, A1N)



## Перечень специальных маркеров

Адрес	Значение	Состояние	Описание	Процессор
M9049	Количество выводимых знаков	выкл.: вывод до кода нуля вкл.: вывод 16 знаков	Если M9049 не установлен, выводятся все знаки до кода нуля (00 <sub>H</sub> ). При установленном маркере выводятся 16 знаков ASCII.	○ (кроме серии "A", A2C)
● M9050	Условие выполнения команды CHG	выкл.: изменения нет вкл.: изменение	Этот маркер устанавливается в качестве условия выполнения команды CHG для изменения результата обработки (более подробную информацию см. в разделе 7.6.8)	● только для A3
M9051	Подавление команды CHG	выкл.: выполнение возможно вкл.: выполнение не возможно	Выполнение команды CHG возможно только при установленном маркере. Этот маркер необходимо установить перед переключением программы. По окончании программы маркер сбрасывается автоматически.	● только для серии "A", A3N
○ M9052	Переключение команды SEG	выкл.: 7-сегментная индикация вкл.: частичное обновление ввода-вывода	При установленном маркере команда SEG имеет функцию 7-сегментной индикации данных. Если маркер не установлен, при выполнении команды SEG происходит частичное обновление входов и выходов, указанных в команде (более подробную информацию см. в п. 6.7.2 и 7.5.5).	○ (кроме серии "A", A2C)
● M9053	Переключение команды EI/DI	выкл.: условие для обновления канала связи вкл.: условие для команды прерывания	Этот маркер изменяет функцию команды EI/DI. При установленном маркере команда служит в качестве условия выполнения программы прерывания. Если маркер не установлен, команда EI/DI служит в качестве условия обновления сети (более подробную информацию см. в п. 6.6.1 и 7.5.5).	● только для AnS, A2C, AnN
M9054	Маркер режима "STEP RUN"	выкл.: иной режим вкл.: STEP RUN	Этот маркер устанавливается после переключения выключателя с ключом в положение "STEP RUN".	○ (кроме A2C)
M9055	Маркер фиксации состояния	выкл.: не окончена вкл.: окончена	Этот маркер устанавливается после завершения фиксации состояния и сбрасывается с помощью команды сброса.	○ (кроме A1, A1N)
M9056	Запрос P, I для основной программы	выкл.: запрос не происходит вкл.: запрос P, I	Запрос указателей P, I после переключения программы (например, для подпрограммы во время обработки основной программы).	● только для A3, A3N, A3H, A3M, A3A
M9057	Запрос P, I для подпрограмм	выкл.: запрос не происходит вкл.: запрос P, I	Маркеры сбрасываются автоматически по окончании запроса указателей.	
M9060	Ошибка в удаленном специальном модуле	выкл.: нормальное состояние вкл.: ошибка	Этот маркер устанавливается при обнаружении ошибки в удаленном специальном модуле (имеется ошибка коммуникации, если коммуникация не возможна даже после количества попыток повторения, указанного в D9174).  После устранения причины ошибки и автоматического повторного подключения дефектной станции к сети маркер сбрасывается. Без автоматического повторного подключения маркер остается установленным.  Прерывание коммуникации при распознании ошибки не приводит к установке или сбросу маркера.	● только для A2C
M9061	Ошибка коммуникации	выкл.: нормальное состояние вкл.: ошибка	Этот маркер устанавливается, если произошел сбой коммуникации с удаленным модулем. Причиной сбоя коммуникации может быть ошибка в данных инициализации, неисправность соединения или выключенное состояние удаленного модуля.  После устранения причины ошибки и автоматического повторного подключения дефектной станции к сети маркер сбрасывается. Без автоматического повторного подключения маркер остается установленным	● только для A2C
M9065	Маркер пошаговой передачи	выкл.: иная обработка вкл.: пошаговая передача	Маркер установлен во время пошаговой передачи экранных масок на модуль AD57(S1)/AD58. По окончании передачи маркер сбрасывается.	● только для AnA, AnAS, AnU
● M9066	Переключение обработки передачи	выкл.: пакетная передача вкл.: пошаговая передача	Если M9066 установлен, экранные маски передаются на модуль AD57(S1)/AD58 пошагово. Если маркер не установлен, передача происходит в виде пакета.	● только для AnA, AnAS, AnU

## Перечень специальных маркеров

Адрес	Значение	Состояние	Описание	Процессор
M9067	Ошибка в модуле ввода-вывода	выкл.: нормальное состояние вкл.: ошибка	Этот маркер устанавливается при обнаружении ошибки в модуле ввода-вывода (имеется ошибка коммуникации, если коммуникация не возможна даже после количества попыток повторения, указанного в регистре D9174).  После устранения причины ошибки и автоматического повторного подключения дефектной станции к сети маркер сбрасывается. Без автоматического повторного подключения маркер остается установленным.  Прерывание коммуникации при распознании ошибки не приводит к установке или сбросу маркера.	● только для A2C
M9068	Испытательный режим	выкл.: нормальный режим вкл.: проверка соединения	Маркер установлен, если в сети выполняется проверка связи с отдельными модулями ввода-вывода и удаленными специальными модулями. При нормальной сетевой коммуникации маркер не установлен.	● только для A2C
M9069	Обработка после ошибки коммуникации	выкл.: все выходы отключаются вкл.: обычный вывод	С помощью этого маркера можно выбрать обработку выходных сигналов при ошибке коммуникации. Если маркер не установлен, то при обнаружении ошибки коммуникации все выходы выключаются. Если маркер установлен, выходы сохраняют фактическое состояние, которое они имели на момент ошибки.	● только для A2C
M9081	Коммуникационный запрос на удаленный специальный модуль	выкл.: запрос возможен вкл.: запрос не возможен	Маркер показывает, возможен ли коммуникационный запрос для удаленного специального модуля (соединенного с модулем Computer-Link или процессором A2C).	● только для A2C, AnA, AnU
M9082	Соответствие установленных номеров станций фактической ситуации	выкл.: соответствуют вкл.: не соответствуют	Этот маркер устанавливается, если последний номер станции подключенного к сети удаленного модуля не совпадает с количеством станций, заданным при инициализации.  Сброс маркера происходит после переключения со STOP на RUN, если заданное при инициализации количество станций совпадает с количеством станций, действительно имеющихся в сети.	● только для A2C
M9084	Контроль на ошибки	выкл.: контроль на ошибки выполняется вкл.: без контроля на ошибки	После обработки команды END центральный процессор проверяет исправность всех предохранителей, сверяет модули ввода-вывода, а также проверяет напряжение батареи буферного питания. В случае ошибки или неисправности выводится сообщение об ошибке. С помощью маркера M9084 контроль на ошибки можно выключить, чтобы уменьшить время обработки команды END.	○ (кроме A2C, AnA, AnAS, AnU)
M9086	Маркер режима RUN для программы BASIC	выкл.: программа BASIC не обрабатывается вкл.: программа BASIC обрабатывается	Этот маркер установлен во время отработки программы A3M-BASIC и сбрасывается по окончании отработки.	● только для A3M
M9087	Маркер состояния PAUSE для программы BASIC	выкл.: отработка программы возможна вкл.: отработка программы не возможна	Этот маркер указывает, возможна ли дальнейшая обработка программы A3M-BASIC, если центральный процессор находится в состоянии PAUSE.  Если маркер не установлен, программа BASIC продолжает обрабатываться и в состоянии PAUSE. Если маркер установлен, отработка программы BASIC также прерывается.	● только для A3M
M9089	Вывод на клемму "ERR"	выкл.: на выходе "ERR" сигнала нет вкл.: сигнал на выходе "ERR"	Этот маркер устанавливается, если основная программа выводит сигнал через клеммы "ERR".  Сброс маркера возможен только при одновременном сбросе маркеров M9089 и M9090.	● только для A2C
M9090	Вывод на клемму "ERR"	выкл.: на выходе "ERR" сигнала нет вкл.: сигнал на выходе "ERR"	Этот маркер устанавливается при возникновении ошибки в MELSECNET/MINI или в основной программе (при останове обработки).  Сброс происходит после устранения ошибки в сети или восстановления основной программы.	● только для A2C
● M9091	Флаг подробной информации об ошибке	выкл.: ошибок нет вкл.: ошибка	В некоторых случаях после возникновения ошибки в регистр данных D9091 записывается более подробный код ошибки и устанавливается маркер M9091. Маркер остается установленным и после устранения причины ошибки.	● только для AnAs, AnU
	Маркер ошибки вызова подпрограммы из области микрокомпьютера	выкл.: ошибок нет вкл.: ошибка	M9091 устанавливается, если во время обработки программы микрокомпьютера возникла ошибка. Маркер остается установленным и после устранения причины ошибки.	○ (кроме AnS, AnAs, AnU)

## Перечень специальных маркеров

Адрес	Значение	Состояние	Описание	Процессор
②③ M9094	Маркер изменения модулей ввода-вывода	выкл.: имеется изменение вкл.: изменений нет	После указания головного адреса для определенного модуля ввода-вывода в регистре D9094 этот модуль можно переключить онлайн путем установки маркера M9094 (только один модуль на каждое выполнение). Изменить рабочее состояние модуля ввода-вывода с помощью маркера M9094 возможно и во время испытательного режима с помощью программатора или при остановленном состоянии центрального процессора. До завершения переключения модуля ввода-вывода в режим онлайн нельзя изменять рабочее состояние центрального процессора.	○ (кроме серии "А", AnS, A3H)

**ПРИМЕЧАНИЕ** После выключения напряжения питания, стирания промежуточной памяти или сброса (RESET) все специальные маркеры сбрасываются.

Переключение выключателя с ключом в положение "STOP" не вызывает сброса маркеров. Фактические состояния сохраняются.

Специальные маркеры, обозначенные символом ①, остаются установленными и после восстановления нормального состояния путем устранения причины ошибки. Сбросить эти специальные маркеры можно следующим образом:

- вставить в основную программу строки, в которых специальный маркер сбрасывается с помощью команды RST при определенном входном условии;
- сбросить их с помощью программатора;
- выполнить сброс центрального процессора путем переключения выключателя с ключом на модуле центрального процессора в положение "RESET".

Специальные маркеры, обозначенные символом ②, устанавливаются и сбрасываются только основной программой.

Специальные маркеры, обозначенные символом ③, устанавливаются и сбрасываются в тестовом режиме с помощью программатора.

### А.4.3 Обзор специальных маркеров связи (только серия "А")

Эти специальные маркеры (маркеры связи) устанавливаются и сбрасываются в зависимости от различных факторов во время обмена данными через коммуникационную сеть. Их состояние изменяется после возникновения ошибки в отработке программы.

Обработка специальных маркеров связи зависит от того, в какой станции находится центральный процессор - мастер-станции или локальной станции.

#### Специальные маркеры связи в мастер-станции

Адрес	Название	Значение	Описание
M9200	Прием команды LRDP	выкл.: не принята вкл.: принята	Этот маркер устанавливается в зависимости от приема команды LRDP (считывание словного операнда). Его можно использовать в программе для блокировки этой команды. Для сброса маркера следует использовать команду RST.
M9201	Обработка команды LRDP	выкл.: не завершена вкл.: завершена	Этот маркер устанавливается в зависимости от состояния обработки команды LRDP (считывание словного операнда). Его можно использовать в программе в качестве входного условия для сброса специальных маркеров M9200 и M9201. Для сброса маркера следует использовать команду RST.
M9202	Прием команды LWTP	выкл.: не принята вкл.: принята	Этот маркер устанавливается в зависимости от приема команды LWTP (запись словного операнда). Его можно использовать в программе для блокировки этой команды. Для сброса маркера следует использовать команду RST.
M9203	Обработка команды LWTP	выкл.: не завершена вкл.: завершена	Этот маркер устанавливается в зависимости от состояния обработки команды LWTP (запись словного операнда). Его можно использовать в программе в качестве входного условия для сброса специальных маркеров M9202 и M9203. Для сброса маркера следует использовать команду RST.
M9206	Ошибка в параметрах связи хост-станции	выкл.: норм. состояние вкл.: ошибка	В зависимости от настройки параметров связи в хост-станции.
M9207	Соответствие параметров связи нескольких мастер-станций	выкл.: норм. состояние вкл.: не соответствуют	В зависимости от соответствия параметров связи (операнды В и W) главной станции 2-го яруса с мастер-станцией 3-го яруса (только в случае сети с 3 ярусами).
M9208	Область передачи В и W для мастер-станции на нижнем ярусе	выкл.: на 2-й и 3-й ярус вкл.: только на 2-й ярус	Этим маркером определяется, требуется ли данные связи, находящиеся в операндах В и W мастер-станции 1-го яруса, передавать в станции нижних ярусов (подстанции). Передача в подстанции происходит только при не установленном маркере M9208.
M9209	Проверка параметров связи (только для мастер-станции нижнего яруса)	выкл.: проверка вкл.: без проверки	Этот специальный маркер устанавливается, если операнды связи (В и W) из верхнего яруса не требуется сравнивать с операндами связи (В и W) нижнего яруса. Если M9209 не установлен, параметры связи верхнего и нижнего яруса проверяются постоянно.
M9210	Неисправность карты связи мастер-станции	выкл.: норм. состояние вкл.: ошибка	В зависимости от аппаратной неисправности или отсутствия карты связи MELSECNET(/В). Обработку выполняет центральный процессор.
M9224	Состояние связи	выкл.: онлайн вкл.: офлайн	В режиме офлайн, при тестировании "от станции к станции" и при выполнении самодиагностики с помощью обратной петли этот маркер установлен.
M9225*	Ошибка в прямой петле	выкл.: норм. состояние вкл.: ошибка	В зависимости от состояния ошибки в прямой петле.
M9226*	Ошибка в обратной петле	выкл.: норм. состояние вкл.: ошибка	В зависимости от состояния ошибки в обратной петле.
M9227*	Состояние тестирования петли	выкл.: тестирование прямой или обратной петли вкл.: тестирование не происходит	В зависимости от выполнения контроля в передней или обратной петле.
M9232	Рабочее состояние локальной станции	выкл.: RUN или STEPRUN вкл.: STOP или PAUSE	В зависимости от рабочего состояния локальной станции (RUN, STOP или PAUSE).

## Специальные маркеры связи в мастер-станции

Адрес	Название	Значение	Описание
M9233	Распознавание ошибки для локальной станции	выкл.: норм. состояние вкл.: ошибка	Зависит от того, распознала ли локальная станция ошибку в другой станции.
M9235	Ошибка параметров в локальной или удаленной станции ввода-вывода	выкл.: норм. состояние вкл.: ошибка	Зависит от обнаружения ошибки в параметрах связи локальной или удаленной станции ввода-вывода.
M9236	Состояние инициализации локальной или удаленной станции ввода-вывода	выкл.: передача не происходит вкл.: передача данных	Зависит от передачи данных инициализации (например, параметров) между мастер-станцией и локальной или удаленной станцией ввода-вывода.
M9237	Ошибка в локальной или удаленной станции ввода-вывода	выкл.: норм. состояние вкл.: ошибка	Зависит от состояния ошибки в локальной или удаленной станции ввода-вывода.
M9238*	Ошибка в петле локальной или удаленной станции ввода-вывода	выкл.: норм. состояние вкл.: ошибка	Зависит от состояния ошибки в прямой или обратной петле локальной или удаленной станции ввода-вывода.

\* Помеченные этой звездочкой специальные маркеры не могут применяться в сети MELSECNET/B.

## Специальные маркеры связи в локальной станции

Адрес	Значение	Значение	Описание
M9204	Обработка команды LRDP	выкл.: не завершена вкл.: завершена	Этот маркер устанавливается в зависимости от состояния обработки команды LRDP в локальной станции.
M9205	Обработка команды LWTP	выкл.: не завершена вкл.: завершена	Этот маркер устанавливается в зависимости от состояния обработки команды LWTP в локальной станции.
M9211	Неисправность карты связи в локальной станции	выкл.: нормальное состояние вкл.: ошибка	Зависит от аппаратной неисправности или отсутствия карты связи MELSECNET(/B). Обработку выполняет центральный процессор.
M9240*	Состояние связи	выкл.: онлайн вкл.: офлайн	В режиме офлайн, при тестировании "от станции к станции" и при выполнении самодиагностики с помощью обратной петли этот маркер установлен.
M9241*	Ошибка в прямой петле	выкл.: нормальное состояние вкл.: ошибка	В зависимости от состояния ошибки в прямой петле.
M9242*	Ошибка в обратной петле	выкл.: нормальное состояние вкл.: ошибка	В зависимости от состояния ошибки в обратной петле.
M9243	Передача по обратной петле	выкл.: не выполняется вкл.: выполняется	Этот специальный маркер установлен, если станция сама выполняет передачу данных по обратной петле.
M9246	Состояние приема данных	выкл.: данные приняты вкл.: данные не приняты	Зависит от передачи данных между мастер-станцией и локальной станцией.
M9247	Состояние приема данных	выкл.: данные приняты вкл.: данные не приняты	Зависит от передачи данных между мастер-станцией определенного яруса и локальной станцией.
M9250	Состояние приема параметров	выкл.: параметры приняты вкл.: параметры не приняты	Зависит от состояния передачи данных параметров из мастер-станции.
M9251	Обрыв передачи	выкл.: нормальное состояние вкл.: обрыв	Зависит от состояния передачи в локальной станции.
M9252	Состояние тестирования петли	выкл.: тестирование не происходит вкл.: тестирование прямой или обратной петли	Зависит от выполнения контроля в передней или обратной петле.
M9253	Рабочее состояние мастер-станции	выкл.: RUN или STEP RUN вкл.: STOP или PAUSE	Зависит от рабочего состояния (RUN, STOP или PAUSE) мастер-станции.
M9254	Рабочее состояние другой локальной станции	выкл.: RUN или STEP RUN вкл.: STOP или PAUSE	Зависит от рабочего состояния (RUN, STOP или PAUSE) локальной станции.
M9255	Распознавание ошибки для других локальных станций	выкл.: нормальное состояние вкл.: ошибка	Зависит от того, распознала ли локальная станция ошибку в другой станции.

\* Помеченные этой звездочкой специальные маркеры не могут применяться в сети MELSEC NET/B.

## A.5 Обзор специальных регистров

### A.5.1 Обзор регистров диагностики (MELSEC серии "Q" и "System Q")

Регистры диагностики SD – это внутренние регистры определенного назначения в программируемом контроллере.

Поэтому эти регистры нельзя использовать в основных программах аналогично обычным регистрам. Однако для управления центральным процессором запись данных в эти регистры возможна. Данные в регистрах диагностики сохраняются в двоичном формате, если только не потребовался иной формат.

В этой таблице разъясняются заголовки столбцов, используемые в таблицах на следующих страницах.

Заголовок таблицы	Значение
Адрес	● адрес регистра диагностики
Название	● название регистра диагностики
Значение	● краткое разъяснение значения регистра
Описание	● подробная информация о значении регистра
Устанавливает (если установлен)	Регистры может устанавливать пользователь или система. <Устанавливает> С : устанавливает система П : устанавливает пользователь (с помощью основной программы или вручную в тестовом режиме периферийного устройства) С/П : устанавливает система и пользователь  Отображается только в случае, если регистр устанавливается системой. <если установлен> Обработка команды END : устанавливается при каждой обработке команды END Инициализация : устанавливается только во время инициализации (при включении сетевого блока или переключении центр. процессора из режима STOP в режим RUN) Изменение состояния : устанавливается только после изменения состояния Ошибка : устанавливается только после возникновения ошибки Выполнение команды : устанавливается при выполнении команды Запрос : устанавливается только по запросу пользователя (с помощью маркера SM и т. п.)
Соответствующий регистр центрального процессора серии "А" D9 [ ] [ ]	● Специальный регистр D9 [ ] [ ], соответствующий процессору серии "А". (изменение, если таковое имеется, и примечание к нему) ● Если в этом столбце указано "новый", то это означает, что этот регистр впервые появился в центральных процессорах серий Q и QnA.
Действ. для:	● Указывает, для какого центрального процессора предназначен этот спец. регистр. ● : действителен для центральных процессоров всех типов Q: действителен только для центральных процессоров MELSEC "System Q" QnA: действителен для центральных процессоров MELSEC серий QnA и Q2AS Тип центрального процессора: действителен только для этого центрального процессора (например, Q4AR) Rem: действителен для удаленных модулей ввода-вывода MELSECNET/H

**ПРИМЕЧАНИЯ** Подробную информацию на следующие темы вы найдете в следующих руководствах:

- Сети → Melsecnet/10/10H/25H Network System Reference Manual for QnA
- Язык SFC → Q-/QnA-CPU Programming Manual ( SFC )

Специальные регистры с SD1200 по SD1255 используются в процессорах серии QnA. В процессорах MELSEC серии "System Q" эти регистры не используются. Специальные регистры, начиная с SD1500, зарезервированы для процессора Q4AR.

Перечень регистров диагностики

(1) Информация для диагностики

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [1][1]	Дейст. для:						
SD0	Ошибка, выявленная при диагностике	Код ошибки, выявленной путем диагностики	<ul style="list-style-type: none"> <li>Код ошибки, выявленной с помощью функции диагностики, сохраняется в двоичном формате.</li> <li>Содержимое этого регистра относится к последнему событию возникновения ошибки.</li> </ul>	С (ошибка)	D9008 изменился формат							
SD1	Время суток, в которое при диагностике была выявлена ошибка	Время суток, в которое при диагностике была выявлена ошибка	<ul style="list-style-type: none"> <li>Год (последние два разряда) и месяц, в которые было обновлено содержимого регистра SD0. Данные сохраняются в двузначной двоично-десятичной кодировке. Пример: <b>октябрь 1995 = 9510</b></li> </ul> <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 33%;">b15</td> <td style="width: 33%;">b8 b7</td> <td style="width: 33%;">b0</td> </tr> <tr> <td>год (от 0 до 99)</td> <td>месяц (от 1 до 12)</td> <td></td> </tr> </table>	b15	b8 b7	b0	год (от 0 до 99)	месяц (от 1 до 12)		С (ошибка)	новый	
b15			b8 b7	b0								
год (от 0 до 99)			месяц (от 1 до 12)									
<ul style="list-style-type: none"> <li>День и час обновления данных в SD0. Данные сохраняются в двузначной двоичнодесятичной кодировке. Пример: <b>25-е число, 22 часа = 2522</b></li> </ul> <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 33%;">b15</td> <td style="width: 33%;">b8 b7</td> <td style="width: 33%;">b0</td> </tr> <tr> <td>день (от 1 до 31)</td> <td>час (от 0 до 23)</td> <td></td> </tr> </table>	b15	b8 b7	b0	день (от 1 до 31)	час (от 0 до 23)							
b15	b8 b7	b0										
день (от 1 до 31)	час (от 0 до 23)											
<ul style="list-style-type: none"> <li>Минута и секунда обновления данных в SD0. Данные сохраняются в двузначной двоичнодесятичной кодировке. Пример: <b>35 мин 48 с = 3548</b></li> </ul> <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 33%;">b15</td> <td style="width: 33%;">b8 b7</td> <td style="width: 33%;">b0</td> </tr> <tr> <td>минута (от 0 до 59)</td> <td>секунда (от 0 до 59)</td> <td></td> </tr> </table>	b15	b8 b7	b0	минута (от 0 до 59)	секунда (от 0 до 59)							
b15	b8 b7	b0										
минута (от 0 до 59)	секунда (от 0 до 59)											
SD3												
SD4	Категории информации об ошибках	Коды категорий информации об ошибках	<p>С помощью кодов категорий можно определить, какого типа информация сохранена в области общей информации об ошибке (SD5–SD15) и области специфической информации об ошибке (SD16–SD26).</p> <table border="1" style="width: 100%; text-align: center;"> <tr> <td style="width: 33%;">b15</td> <td style="width: 33%;">b8 b7</td> <td style="width: 33%;">b0</td> </tr> <tr> <td>Специфическая информация об ошибках</td> <td>Общая информация об ошибках</td> <td></td> </tr> </table> <ul style="list-style-type: none"> <li>Коды категорий общей информации об ошибках сохраняются следующим образом:                     <ol style="list-style-type: none"> <li>ошибок нет</li> <li>номер станции/модуля/центр. проц./монт. шины</li> <li>название файла/дисковода</li> <li>время (установленное значение)</li> <li>локализация ошибки программы</li> <li>основание переключения (только в случае проц. Q4AR)</li> </ol> </li> <li>Коды категорий специфической информации об ошибках сохраняются следующим образом:                     <ol style="list-style-type: none"> <li>ошибок нет</li> <li>(открыто)</li> <li>название файла/дисковода</li> <li>время (фактическое измеренное значение)</li> <li>локализация ошибки программы</li> <li>номер параметра</li> <li>номер маркера ошибки</li> <li>номер ошибки команды CHK</li> </ol> </li> </ul>	b15	b8 b7	b0	Специфическая информация об ошибках	Общая информация об ошибках		С (ошибка)	новый	Rem
b15	b8 b7	b0										
Специфическая информация об ошибках	Общая информация об ошибках											



Перечень регистров диагностики (продолжение)

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:																																																																																																							
SD5	Общая информация об ошибках		<ul style="list-style-type: none"> <li>Здесь сохраняется общая информация, относящаяся к кодам ошибок (SD0).</li> <li>Сохраняется информация следующих 5 видов:                             <ol style="list-style-type: none"> <li>номер станции/модуля                                     <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr><td>SD5</td><td>номер станции/модуля</td></tr> <tr><td>SD6</td><td>номер входа или выхода</td></tr> <tr><td>SD7</td><td rowspan="10">свободно</td></tr> <tr><td>SD8</td></tr> <tr><td>SD9</td></tr> <tr><td>SD10</td></tr> <tr><td>SD11</td></tr> <tr><td>SD12</td></tr> <tr><td>SD13</td></tr> <tr><td>SD14</td></tr> <tr><td>SD15</td></tr> </tbody> </table> </li> <li>Название файла/дисковода                                     <p style="margin-left: 20px;">Пример: название файла = ABCDEFGH.IJK</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr><td>SD5</td><td>дисковод</td></tr> <tr><td>SD6</td><td rowspan="4">название файла (ASCII-код: 8 знаков)</td></tr> <tr><td>SD7</td></tr> <tr><td>SD8</td></tr> <tr><td>SD9</td></tr> <tr> <td>SD10</td> <td>расширение 2Eн(.)</td> </tr> <tr> <td>SD11</td> <td>(ASCII-код: 3 знака)</td> </tr> <tr><td>SD12</td><td rowspan="4">свободно</td></tr> <tr><td>SD13</td></tr> <tr><td>SD14</td></tr> <tr><td>SD15</td></tr> </tbody> </table> </li> <li>Время (установленное значение)                                     <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr><td>SD5</td><td>время: с шагом в 1 мкс (0–999 мкс)</td></tr> <tr><td>SD6</td><td>время: с шагом в 1 мс (0–65535 мс)</td></tr> <tr><td>SD7</td><td rowspan="10">свободно</td></tr> <tr><td>SD8</td></tr> <tr><td>SD9</td></tr> <tr><td>SD10</td></tr> <tr><td>SD11</td></tr> <tr><td>SD12</td></tr> <tr><td>SD13</td></tr> <tr><td>SD14</td></tr> <tr><td>SD15</td></tr> </tbody> </table> </li> <li>Локализация ошибки программы                                     <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr><td>SD5</td><td rowspan="4">название файла (ASCII-код: 8 знаков)</td></tr> <tr><td>SD6</td></tr> <tr><td>SD7</td></tr> <tr><td>SD8</td></tr> <tr> <td>SD9</td> <td>расширение 2Eн (.)</td> </tr> <tr> <td>SD10</td> <td>(ASCII-код: 3 знака)</td> </tr> <tr><td>SD11</td><td>конфигурация битов *</td></tr> <tr><td>SD12</td><td>№ блока</td></tr> <tr><td>SD13</td><td>№ шага/перехода</td></tr> <tr><td>SD14</td><td>№ шага программы (L)</td></tr> <tr><td>SD15</td><td>№ шага программы (H)</td></tr> </tbody> </table> <p style="margin-left: 20px;">* Конфигурация битов:</p> <table border="1" style="margin-left: 20px;"> <tr> <td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p style="margin-left: 20px;">← (№ бита)</p> <p style="margin-left: 20px;">не используется</p> <p style="margin-left: 20px;">                     блок CFS имеется (1)/не имеется (0)                      шаг CFS имеется (1)/не имеется (0)                      переход CFS имеется (1)/не имеется (0)                 </p> </li> </ol> </li> </ul>	Номер	Значение	SD5	номер станции/модуля	SD6	номер входа или выхода	SD7	свободно	SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15	Номер	Значение	SD5	дисковод	SD6	название файла (ASCII-код: 8 знаков)	SD7	SD8	SD9	SD10	расширение 2Eн(.)	SD11	(ASCII-код: 3 знака)	SD12	свободно	SD13	SD14	SD15	Номер	Значение	SD5	время: с шагом в 1 мкс (0–999 мкс)	SD6	время: с шагом в 1 мс (0–65535 мс)	SD7	свободно	SD8	SD9	SD10	SD11	SD12	SD13	SD14	SD15	Номер	Значение	SD5	название файла (ASCII-код: 8 знаков)	SD6	SD7	SD8	SD9	расширение 2Eн (.)	SD10	(ASCII-код: 3 знака)	SD11	конфигурация битов *	SD12	№ блока	SD13	№ шага/перехода	SD14	№ шага программы (L)	SD15	№ шага программы (H)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	С (ошибка)	новый	●
Номер				Значение																																																																																																									
SD5				номер станции/модуля																																																																																																									
SD6				номер входа или выхода																																																																																																									
SD7				свободно																																																																																																									
SD8																																																																																																													
SD9																																																																																																													
SD10																																																																																																													
SD11																																																																																																													
SD12																																																																																																													
SD13																																																																																																													
SD14																																																																																																													
SD15																																																																																																													
Номер					Значение																																																																																																								
SD5				дисковод																																																																																																									
SD6	название файла (ASCII-код: 8 знаков)																																																																																																												
SD7																																																																																																													
SD8																																																																																																													
SD9																																																																																																													
SD10	расширение 2Eн(.)																																																																																																												
SD11	(ASCII-код: 3 знака)																																																																																																												
SD12	свободно																																																																																																												
SD13																																																																																																													
SD14																																																																																																													
SD15																																																																																																													
Номер	Значение																																																																																																												
SD5	время: с шагом в 1 мкс (0–999 мкс)																																																																																																												
SD6	время: с шагом в 1 мс (0–65535 мс)																																																																																																												
SD7	свободно																																																																																																												
SD8																																																																																																													
SD9																																																																																																													
SD10																																																																																																													
SD11																																																																																																													
SD12																																																																																																													
SD13																																																																																																													
SD14																																																																																																													
SD15																																																																																																													
Номер		Значение																																																																																																											
SD5	название файла (ASCII-код: 8 знаков)																																																																																																												
SD6																																																																																																													
SD7																																																																																																													
SD8																																																																																																													
SD9	расширение 2Eн (.)																																																																																																												
SD10	(ASCII-код: 3 знака)																																																																																																												
SD11	конфигурация битов *																																																																																																												
SD12	№ блока																																																																																																												
SD13	№ шага/перехода																																																																																																												
SD14	№ шага программы (L)																																																																																																												
SD15	№ шага программы (H)																																																																																																												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																																														
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																																																																														

## Перечень регистров диагностики (продолжение)

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:																																																																																		
Значение расширений названия файлов:																																																																																								
<table border="1"> <thead> <tr> <th colspan="2">SD10 (SD9)</th> <th colspan="2">SD11 (SD10)</th> <th rowspan="2">Расширение</th> <th rowspan="2">Тип файла</th> </tr> <tr> <th>Старший байт</th> <th>Младший байт</th> <th>Младший байт</th> <th>Старший байт</th> </tr> </thead> <tbody> <tr> <td>51H</td> <td>50H</td> <td>50H</td> <td>41H</td> <td>QPA</td> <td>параметры</td> </tr> <tr> <td>51H</td> <td>50H</td> <td>50H</td> <td>47H</td> <td>QPG</td> <td>программы</td> </tr> <tr> <td>51H</td> <td>43H</td> <td>44H</td> <td>44H</td> <td>QCD</td> <td>комментарии к операндам</td> </tr> <tr> <td>51H</td> <td>44H</td> <td>49H</td> <td>49H</td> <td>QDI</td> <td>начальные значения операндов</td> </tr> <tr> <td>51H</td> <td>44H</td> <td>52H</td> <td>52H</td> <td>QDR</td> <td>регистры файлов</td> </tr> <tr> <td>51H</td> <td>44H</td> <td>53H</td> <td>53H</td> <td>QDS</td> <td>данные имитации</td> </tr> <tr> <td>51H</td> <td>44H</td> <td>4CH</td> <td>4CH</td> <td>QDL</td> <td>локальные операнды</td> </tr> <tr> <td>51H</td> <td>54H</td> <td>53H</td> <td>53H</td> <td>QTS</td> <td>данные выб. трассировки (только QnA)</td> </tr> <tr> <td>51H</td> <td>54H</td> <td>4CH</td> <td>4CH</td> <td>QTL</td> <td>данные фиксации состояния (только QnA)</td> </tr> <tr> <td>51H</td> <td>54H</td> <td>50H</td> <td>50H</td> <td>QTP</td> <td>данные трассировки программы (QnA)</td> </tr> <tr> <td>51H</td> <td>54H</td> <td>52H</td> <td>52H</td> <td>QTR</td> <td>файл трассировки для прогр. на языке SFC</td> </tr> <tr> <td>51H</td> <td>46H</td> <td>44H</td> <td>44H</td> <td>QFD</td> <td>данные ошибок</td> </tr> </tbody> </table>							SD10 (SD9)		SD11 (SD10)		Расширение	Тип файла	Старший байт	Младший байт	Младший байт	Старший байт	51H	50H	50H	41H	QPA	параметры	51H	50H	50H	47H	QPG	программы	51H	43H	44H	44H	QCD	комментарии к операндам	51H	44H	49H	49H	QDI	начальные значения операндов	51H	44H	52H	52H	QDR	регистры файлов	51H	44H	53H	53H	QDS	данные имитации	51H	44H	4CH	4CH	QDL	локальные операнды	51H	54H	53H	53H	QTS	данные выб. трассировки (только QnA)	51H	54H	4CH	4CH	QTL	данные фиксации состояния (только QnA)	51H	54H	50H	50H	QTP	данные трассировки программы (QnA)	51H	54H	52H	52H	QTR	файл трассировки для прогр. на языке SFC	51H	46H	44H	44H	QFD	данные ошибок
SD10 (SD9)		SD11 (SD10)		Расширение	Тип файла																																																																																			
Старший байт	Младший байт	Младший байт	Старший байт																																																																																					
51H	50H	50H	41H	QPA	параметры																																																																																			
51H	50H	50H	47H	QPG	программы																																																																																			
51H	43H	44H	44H	QCD	комментарии к операндам																																																																																			
51H	44H	49H	49H	QDI	начальные значения операндов																																																																																			
51H	44H	52H	52H	QDR	регистры файлов																																																																																			
51H	44H	53H	53H	QDS	данные имитации																																																																																			
51H	44H	4CH	4CH	QDL	локальные операнды																																																																																			
51H	54H	53H	53H	QTS	данные выб. трассировки (только QnA)																																																																																			
51H	54H	4CH	4CH	QTL	данные фиксации состояния (только QnA)																																																																																			
51H	54H	50H	50H	QTP	данные трассировки программы (QnA)																																																																																			
51H	54H	52H	52H	QTR	файл трассировки для прогр. на языке SFC																																																																																			
51H	46H	44H	44H	QFD	данные ошибок																																																																																			

Перечень регистров диагностики (продолжение)

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:																																																					
SD16	Общая информация об ошибках	Здесь сохраняется общая информация, относящаяся с кодам ошибок (SD0). Здесь сохраняются следующие 6 видов данных:	<p>(1) название файла/дисковода      Пример: <b>название файла = ABCDEFGH.IJK</b></p> <table border="1"> <thead> <tr> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr> <td>SD16</td> <td>дисковод</td> </tr> <tr> <td>SD17</td> <td rowspan="4">название файла (ASCII-код: 8 знаков)</td> </tr> <tr> <td>SD18</td> </tr> <tr> <td>SD19</td> </tr> <tr> <td>SD20</td> </tr> <tr> <td>SD21</td> <td>расширение      2Eн(.)</td> </tr> <tr> <td>SD22</td> <td>(ASCII-код: 3 знака)</td> </tr> <tr> <td>SD23</td> <td rowspan="4">свободно</td> </tr> <tr> <td>SD24</td> </tr> <tr> <td>SD25</td> </tr> <tr> <td>SD26</td> </tr> </tbody> </table> <p>b15      b0</p> <table border="1"> <tr> <td>B</td> <td>A</td> </tr> <tr> <td>D</td> <td>C</td> </tr> <tr> <td>F</td> <td>E</td> </tr> <tr> <td>H</td> <td>G</td> </tr> <tr> <td>I</td> <td>.</td> </tr> <tr> <td>K</td> <td>J</td> </tr> </table>	Номер	Значение	SD16	дисковод	SD17	название файла (ASCII-код: 8 знаков)	SD18	SD19	SD20	SD21	расширение      2Eн(.)	SD22	(ASCII-код: 3 знака)	SD23	свободно	SD24	SD25	SD26	B	A	D	C	F	E	H	G	I	.	K	J	C (ошибка)	новый	●																							
Номер				Значение																																																							
SD16				дисковод																																																							
SD17				название файла (ASCII-код: 8 знаков)																																																							
SD18																																																											
SD19																																																											
SD20																																																											
SD21				расширение      2Eн(.)																																																							
SD22				(ASCII-код: 3 знака)																																																							
SD23				свободно																																																							
SD24																																																											
SD25																																																											
SD26																																																											
B	A																																																										
D	C																																																										
F	E																																																										
H	G																																																										
I	.																																																										
K	J																																																										
SD17																																																											
SD18																																																											
SD19																																																											
SD20																																																											
SD21																																																											
SD22																																																											
SD23																																																											
SD24																																																											
SD25																																																											
SD26																																																											
SD16	Общая информация об ошибках	Здесь сохраняется общая информация, относящаяся с кодам ошибок (SD0). Здесь сохраняются следующие 6 видов данных:	<p>(2) Время (установленное значение)</p> <table border="1"> <thead> <tr> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr> <td>SD16</td> <td>время: с шагом 1 мкс (0–999 мкс)</td> </tr> <tr> <td>SD17</td> <td>время: с шагом 1 мс (0–65535 мс)</td> </tr> <tr> <td>SD18</td> <td rowspan="10">свободно</td> </tr> <tr> <td>SD19</td> </tr> <tr> <td>SD20</td> </tr> <tr> <td>SD21</td> </tr> <tr> <td>SD22</td> </tr> <tr> <td>SD23</td> </tr> <tr> <td>SD24</td> </tr> <tr> <td>SD25</td> </tr> <tr> <td>SD26</td> </tr> </tbody> </table>	Номер	Значение	SD16	время: с шагом 1 мкс (0–999 мкс)	SD17	время: с шагом 1 мс (0–65535 мс)	SD18	свободно	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26	C (ошибка)	новый	●																																					
Номер				Значение																																																							
SD16				время: с шагом 1 мкс (0–999 мкс)																																																							
SD17				время: с шагом 1 мс (0–65535 мс)																																																							
SD18				свободно																																																							
SD19																																																											
SD20																																																											
SD21																																																											
SD22																																																											
SD23																																																											
SD24																																																											
SD25																																																											
SD26																																																											
SD17																																																											
SD18																																																											
SD19																																																											
SD20																																																											
SD21																																																											
SD22																																																											
SD23																																																											
SD24																																																											
SD25																																																											
SD26																																																											
SD16	Общая информация об ошибках	Здесь сохраняется общая информация, относящаяся с кодам ошибок (SD0). Здесь сохраняются следующие 6 видов данных:	<p>(3) Локализация ошибки программы</p> <table border="1"> <thead> <tr> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr> <td>SD16</td> <td rowspan="4">название файла (ASCII-код: 8 знаков)</td> </tr> <tr> <td>SD17</td> </tr> <tr> <td>SD18</td> </tr> <tr> <td>SD19</td> </tr> <tr> <td>SD20</td> <td>расширение      2Eн(.)</td> </tr> <tr> <td>SD21</td> <td>(ASCII-код: 3 знака)</td> </tr> <tr> <td>SD22</td> <td>конфигурация битов *</td> </tr> <tr> <td>SD23</td> <td>№ блока</td> </tr> <tr> <td>SD24</td> <td>№ шага/перехода</td> </tr> <tr> <td>SD25</td> <td>№ шага программы (L)</td> </tr> <tr> <td>SD26</td> <td>№ шага программы (H)</td> </tr> </tbody> </table> <p>* Конфигурация битов:</p> <table border="1"> <tr> <td>15</td> <td>14</td> <td>13</td> <td>12</td> <td>11</td> <td>10</td> <td>9</td> <td>8</td> <td>7</td> <td>6</td> <td>5</td> <td>4</td> <td>3</td> <td>2</td> <td>1</td> <td>0</td> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> <td>0</td> </tr> </table> <p>← (№ бита)</p> <p>не используется      блок CFS имеется (1)/не имеется (0) шаг CFS имеется (1)/не имеется (0) переход CFS имеется (1)/не имеется (0)</p>	Номер	Значение	SD16	название файла (ASCII-код: 8 знаков)	SD17	SD18	SD19	SD20	расширение      2Eн(.)	SD21	(ASCII-код: 3 знака)	SD22	конфигурация битов *	SD23	№ блока	SD24	№ шага/перехода	SD25	№ шага программы (L)	SD26	№ шага программы (H)	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C (ошибка)	новый	●
Номер				Значение																																																							
SD16				название файла (ASCII-код: 8 знаков)																																																							
SD17																																																											
SD18																																																											
SD19																																																											
SD20				расширение      2Eн(.)																																																							
SD21				(ASCII-код: 3 знака)																																																							
SD22				конфигурация битов *																																																							
SD23				№ блока																																																							
SD24				№ шага/перехода																																																							
SD25				№ шага программы (L)																																																							
SD26	№ шага программы (H)																																																										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																												
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																																												
SD17																																																											
SD18																																																											
SD19																																																											
SD20																																																											
SD21																																																											
SD22																																																											
SD23																																																											
SD24																																																											
SD25																																																											
SD26																																																											

Перечень регистров диагностики (продолжение)

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:																																														
SD16 SD17 SD18 SD19 SD20 SD21 SD22 SD23 SD24 SD25	Специфическая информация об ошибках		(4) № параметра (5) № маркера ош. № ошибки команды ЧНК	С (ошибка)	новый																																															
SD26			<table border="1"> <thead> <tr> <th>Номер</th> <th>Значение</th> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr> <td>SD16</td> <td>№ параметра</td> <td>SD16</td> <td>№ параметра</td> </tr> <tr> <td>SD17</td> <td rowspan="10">не используется</td> <td>SD17</td> <td rowspan="10">не используется</td> </tr> <tr> <td>SD18</td> <td>SD18</td> </tr> <tr> <td>SD19</td> <td>SD19</td> </tr> <tr> <td>SD20</td> <td>SD20</td> </tr> <tr> <td>SD21</td> <td>SD21</td> </tr> <tr> <td>SD22</td> <td>SD22</td> </tr> <tr> <td>SD23</td> <td>SD23</td> </tr> <tr> <td>SD24</td> <td>SD24</td> </tr> <tr> <td>SD25</td> <td>SD25</td> </tr> <tr> <td>SD26</td> <td>SD26</td> </tr> </tbody> </table> <p>(6) Ошибка параметрирования специальных модулей (только в случае процессоров MELSEC "System Q")</p> <table border="1"> <thead> <tr> <th>Номер</th> <th>Значение</th> </tr> </thead> <tbody> <tr> <td>SD16</td> <td>№ параметра</td> </tr> <tr> <td>SD17</td> <td>код ошибки для спец. модуля</td> </tr> <tr> <td>SD18</td> <td rowspan="10">не использ.</td> </tr> <tr> <td>SD19</td> </tr> <tr> <td>SD20</td> </tr> <tr> <td>SD21</td> </tr> <tr> <td>SD22</td> </tr> <tr> <td>SD23</td> </tr> <tr> <td>SD24</td> </tr> <tr> <td>SD25</td> </tr> <tr> <td>SD26</td> </tr> </tbody> </table>				Номер	Значение	Номер	Значение	SD16	№ параметра	SD16	№ параметра	SD17	не используется	SD17	не используется	SD18	SD18	SD19	SD19	SD20	SD20	SD21	SD21	SD22	SD22	SD23	SD23	SD24	SD24	SD25	SD25	SD26	SD26	Номер	Значение	SD16	№ параметра	SD17	код ошибки для спец. модуля	SD18	не использ.	SD19	SD20	SD21	SD22	SD23	SD24	SD25	SD26
Номер			Значение				Номер	Значение																																												
SD16			№ параметра				SD16	№ параметра																																												
SD17			не используется				SD17	не используется																																												
SD18							SD18																																													
SD19							SD19																																													
SD20							SD20																																													
SD21							SD21																																													
SD22							SD22																																													
SD23	SD23																																																			
SD24	SD24																																																			
SD25	SD25																																																			
SD26	SD26																																																			
Номер	Значение																																																			
SD16	№ параметра																																																			
SD17	код ошибки для спец. модуля																																																			
SD18	не использ.																																																			
SD19																																																				
SD20																																																				
SD21																																																				
SD22																																																				
SD23																																																				
SD24																																																				
SD25																																																				
SD26																																																				
SD50		Сброс ошибки	Номер сброшенной ошибки	Сохраняет номер сброшенной ошибки	П	новый																																														
SD51	Слишком низкое напряжение батареи (фиксируемый маркер)	Набор битов, показывающий, где произошло падение напряжения батареи.	<ul style="list-style-type: none"> <li>● Соответствующие биты устанавливаются, если упало напряжение батареи.</li> <li>● Этот бит остается установленным, даже если напряжение батареи снова достигло нормального уровня.</li> </ul> <p>В процессорах Q00J, Q00 и Q01 устанавливается только бит 0.</p>	С (ошибка)	новый																																															
SD52	Низкое напряжение батареи	Набор битов, показывающий, где снизилось напряжение батареи.	<ul style="list-style-type: none"> <li>● Действует аналогично вышеописанному регистру SD51 (см. выше)</li> <li>● Этот бит сбрасывается после того, как напряжение батареи достигло нормального значения.</li> </ul>	С (ошибка)	новый																																															
SD53	Падение напряжения питания	Количество падений напряжения	<ul style="list-style-type: none"> <li>● При каждом падении напряжения во время работы более чем на 20 % от номинального напряжения значение этого регистра повышается на „1“. Значение сохраняется в двоичном виде.</li> </ul>	С (ошибка)	D9005	Rem																																														

Перечень регистров диагностики (продолжение)

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:
SD54	Ошибка связи по MINI	Состояние обнаружения ошибки	<p>(1) Устанавливается бит соответствующей станции, если устанавливается один из головных адресов установленного модуля MINI (-S3)  <math>X(n+0)/X(n+20)</math>, <math>X(n+6)/(n+26)</math>, <math>X(n+7)/(n+27)</math> или <math>X(n+8)/X(n+28)</math>.</p> <p>(2) Устанавливается соответствующий бит, если коммуникация между установленным модулем MINI (-S3) и центральным процессором не возможна.</p>	С (ошибка)	D9004 изменился формат	QnA
SD60	Номер неисправного предохранителя	Номер модуля, в котором неисправен предохранитель.	Сохраненное здесь значение является самым нижним адресом станции модуля, в котором неисправен предохранитель, после деления этого адреса на 16.	С (ошибка)	D9000	● Rem
SD61	Ошибка сверки модуля ввода-вывода	Номер модуля, в котором имеется ошибка сверки.	Самый низкий адрес модуля, в котором распознана первая ошибка сверки.	С (ошибка)	D9002	
SD62	№ маркера ошибки		Здесь сохраняется номер ошибки, обнаруженной первой.	С (выполнение команды)	D9009	●
SD63	Количество маркеров ошибок		Количество маркеров ошибок.	С (выполнение команды)	D9124	

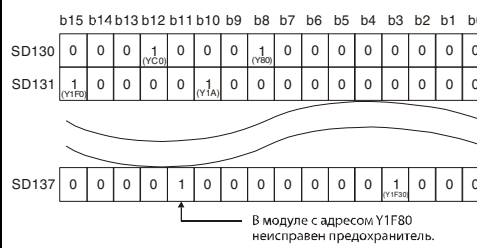
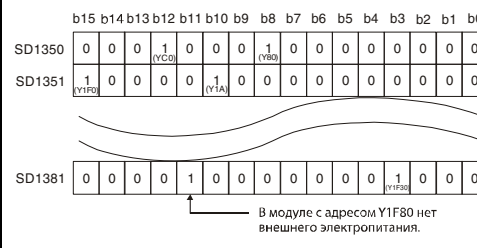
Перечень регистров диагностики (продолжение)

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ][ ][ ]	Дейст. для:																																																																																																																																																																																																																																								
SD64	Таблица номеров обнаруженных маркеров ошибки	Номера обнаруженных маркеров ошибки	<p>Если маркер ошибки устанавливается командой OUT F или SET F, то адрес установленного маркера ошибки записывается в двоичном виде в регистры с SD64 по SD79. Адрес маркера ошибки, сбрасываемый командой RST F, стирается из области регистра. Затем содержимое последующих регистров данных сдвигается вверх на один регистр. При выполнении команды LEDR содержимое регистров SD64...SD79 смещается на один бит вверх. (Этот процесс происходит также при переводе выключателя с ключом на модуле центрального процессора (Q3A/Q4A) в положение RESET.) Если имеется более 16 сообщений об ошибках, 17-й маркер ошибки не сохраняется в регистрах SD64...SD79.</p> <p style="text-align: center;">                     SET SET SET SET SET SET SET SET SET SET SET SET SET SET SET SET                      F50 F25 F19 F25 F15 F70 F65 F38 F110F151F210 LEDR                      → → → → → → → → → → → → → → → →                 </p> <p>SD62 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>99</td></tr></table> <small>распознанные адреса</small></p> <p>SD63 <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>8</td></tr></table> <small>количество обнаруженных маркеров ошибки</small></p> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr><td>SD64</td><td>0</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>99</td></tr> <tr><td>SD65</td><td>0</td><td>0</td><td>25</td><td>25</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>15</td></tr> <tr><td>SD66</td><td>0</td><td>0</td><td>0</td><td>99</td><td>0</td><td>15</td><td>15</td><td>15</td><td>15</td><td>15</td><td>15</td><td>15</td></tr> <tr><td>SD67</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>70</td><td>70</td><td>70</td><td>70</td><td>70</td><td>70</td><td>65</td></tr> <tr><td>SD68</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>65</td><td>65</td><td>65</td><td>65</td><td>38</td><td>38</td></tr> <tr><td>SD69</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>38</td><td>38</td><td>38</td><td>38</td><td>110</td></tr> <tr><td>SD70</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>110</td><td>110</td><td>110</td><td>151</td></tr> <tr><td>SD71</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>151</td><td>151</td><td>210</td></tr> <tr><td>SD72</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>210</td><td>0</td></tr> <tr><td>SD73</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>SD74</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>SD75</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>SD76</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>SD77</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>SD78</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>SD79</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table> <p style="text-align: right;"><small>распознанные адреса</small></p>	0	50	50	50	50	50	50	50	50	50	50	99	0	1	2	3	2	3	4	5	6	7	8	8	SD64	0	50	50	50	50	50	50	50	50	50	50	99	SD65	0	0	25	25	99	99	99	99	99	99	99	15	SD66	0	0	0	99	0	15	15	15	15	15	15	15	SD67	0	0	0	0	0	70	70	70	70	70	70	65	SD68	0	0	0	0	0	0	65	65	65	65	38	38	SD69	0	0	0	0	0	0	0	38	38	38	38	110	SD70	0	0	0	0	0	0	0	0	110	110	110	151	SD71	0	0	0	0	0	0	0	0	0	151	151	210	SD72	0	0	0	0	0	0	0	0	0	0	210	0	SD73	0	0	0	0	0	0	0	0	0	0	0	0	SD74	0	0	0	0	0	0	0	0	0	0	0	0	SD75	0	0	0	0	0	0	0	0	0	0	0	0	SD76	0	0	0	0	0	0	0	0	0	0	0	0	SD77	0	0	0	0	0	0	0	0	0	0	0	0	SD78	0	0	0	0	0	0	0	0	0	0	0	0	SD79	0	0	0	0	0	0	0	0	0	0	0	0	С (выполнение команды)	D9125	●
0				50	50	50	50	50	50	50	50	50	50	99																																																																																																																																																																																																																																
0				1	2	3	2	3	4	5	6	7	8	8																																																																																																																																																																																																																																
SD64				0	50	50	50	50	50	50	50	50	50	50	99																																																																																																																																																																																																																															
SD65				0	0	25	25	99	99	99	99	99	99	99	15																																																																																																																																																																																																																															
SD66				0	0	0	99	0	15	15	15	15	15	15	15																																																																																																																																																																																																																															
SD67				0	0	0	0	0	70	70	70	70	70	70	65																																																																																																																																																																																																																															
SD68				0	0	0	0	0	0	65	65	65	65	38	38																																																																																																																																																																																																																															
SD69				0	0	0	0	0	0	0	38	38	38	38	110																																																																																																																																																																																																																															
SD70				0	0	0	0	0	0	0	0	110	110	110	151																																																																																																																																																																																																																															
SD71				0	0	0	0	0	0	0	0	0	151	151	210																																																																																																																																																																																																																															
SD72				0	0	0	0	0	0	0	0	0	0	210	0																																																																																																																																																																																																																															
SD73				0	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																															
SD74				0	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																															
SD75	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																		
SD76	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																		
SD77	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																		
SD78	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																		
SD79	0	0	0	0	0	0	0	0	0	0	0	0																																																																																																																																																																																																																																		
SD65	D9126																																																																																																																																																																																																																																													
SD66	D9127																																																																																																																																																																																																																																													
SD67	D9128																																																																																																																																																																																																																																													
SD68	D9129																																																																																																																																																																																																																																													
SD69	D9130																																																																																																																																																																																																																																													
SD70	D9131																																																																																																																																																																																																																																													
SD71	D9132																																																																																																																																																																																																																																													
SD72	новый																																																																																																																																																																																																																																													
SD74	новый																																																																																																																																																																																																																																													
SD75	новый																																																																																																																																																																																																																																													
SD76	новый																																																																																																																																																																																																																																													
SD77	новый																																																																																																																																																																																																																																													
SD78	новый																																																																																																																																																																																																																																													
SD79	новый																																																																																																																																																																																																																																													
SD80	Код ошибки команды СНК		Код ошибки, распознанный командой СНК, сохраняется в двоично-десятичном формате.	С (выполнение команды)	новый	● кроме Q00J, Q00 и Q01																																																																																																																																																																																																																																								

Перечень регистров диагностики (продолжение)

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистры проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:										
SD90	Настройка контрольного таймера для контроля шагов и переходов (возможно только при наличии программы на языке SFC)	Номер маркера для ошибки настройки таймера и превышения времени	соответствует SM90	<ul style="list-style-type: none"> <li>● Номер маркера ошибки, устанавливаемого при указании неправильного времени для контрольного таймера или превышения времени контрольного таймера.</li> </ul> <div style="text-align: center;"> <table style="margin: auto;"> <tr> <td style="border: 1px solid black; padding: 2px;">b15</td> <td style="border: 1px solid black; padding: 2px;">b8 b7</td> <td style="border: 1px solid black; padding: 2px;">b0</td> </tr> <tr> <td style="text-align: center;">↑</td> <td style="text-align: center;">↑</td> <td></td> </tr> <tr> <td style="text-align: center; font-size: 8px;">настройка номер F (от 0 до 255)</td> <td style="text-align: center; font-size: 8px;">настройка предельного времени таймера (от 1 до 255 с, с шагом в 1 с)</td> <td></td> </tr> </table> </div> <ul style="list-style-type: none"> <li>● Таймер запускается, если установлены маркеры диагностики SM90...SM99 и шаг активен. Если на протяжении настроенного времени условие дальнейшего переключения соответствующего шага не выполняется, становится маркер ошибки (F).</li> </ul>	b15	b8 b7	b0	↑	↑		настройка номер F (от 0 до 255)	настройка предельного времени таймера (от 1 до 255 с, с шагом в 1 с)		П	D9108	<ul style="list-style-type: none"> <li>● кроме Q00J, Q00 и Q01</li> </ul>
b15			b8 b7		b0											
↑			↑													
настройка номер F (от 0 до 255)			настройка предельного времени таймера (от 1 до 255 с, с шагом в 1 с)													
SD91			соответствует SM91		D9109											
SD92			соответствует SM92		D9110											
SD93			соответствует SM93		D9111											
SD94			соответствует SM94		D9112											
SD95			соответствует SM95		D9113											
SD96			соответствует SM96		D9114											
SD97	соответствует SM97	новый														
SD98	соответствует SM98	новый														
SD99	соответствует SM99	новый														
SD100	Скорость передачи	Память для настроенной скорости передачи последовательного интерфейса	K96: 9600 бит/с, K192: 19.2 кбит/с, K384: 38.4 кбит/с, K576: 57.6 кбит/с, K1152: 115.2 кбит/с	С (при включении напряжения питания или после сброса)	новый	Q00J, Q00 и Q01										
SD101	Настройки коммуникации	Память для настроек последовательной коммуникации	бит 4 = выкл.: без контрольной суммы бит 4 = вкл.: с контрольной суммой бит 5 = выкл.: изменение программы онлайн не допускается бит 5 = вкл.: изменение программы онлайн разрешено Прочие биты не имеют значения.		новый											
SD102	Время ожидания	Память для времени ожидания при последовательной коммуникации	0: без времени ожидания от 1 до F <sub>H</sub> : время ожидания в единицах по 10 мс Предварительная настройка: 0		новый											
SD105	Скорость передачи для CN1 (RS232)	Память для настроенной скорости передачи.	K3: 300 бит/с, K6: 600 бит/с, K24: 2400 бит/с, K48: 4800 бит/с, K96: 9600 бит/с, K192: 19.2 кбит/с, K384: 38.4 кбит/с, K576: 57.6 кбит/с, K1152: 115.2 кбит/с	С	новый	Q кроме Q00J, Q00 и Q01										
SD110	Результат передачи	Код ошибки при передаче данных	Если при передаче данных путем последовательной коммуникации возникла ошибка, то здесь сохраняется код ошибки.	С (ошибка)	новый	Q00J, Q00 и Q01										
SD111	Результат приема	Код ошибки при приеме данных	Если при приеме данных путем последовательной коммуникации возникла ошибка, то здесь сохраняется код ошибки.	С (ошибка)	новый											
SD120	Номер ошибки при исчезновении внешнего напряжения питания	Номер модуля, в котором исчезло внешнее электропитание	Сохраняется самый низкий адрес модуля "System Q", в котором исчезло напряжение питания. (в стадии подготовки)	С (ошибка)	новый	Q кроме Q00J, Q00 и Q01										

Перечень регистров диагностики (продолжение)

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистры проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:
SD130	Модули с неисправным предохранителем	Набор битов (16 бит) показывает модули с неисправным предохранителем 0: Неисправных предохранит. нет 1: Имеется неисправный предохранитель	<ul style="list-style-type: none"> <li>Количество выходных модулей с неисправным предохранителем сохраняется в виде набора из 16 битов. (Если номер модуля установлен в параметрах, то сохраняется этот номер.)</li> <li>Неисправные предохранители распознаются и в модулях выводаудаленных станций.</li> <li>После замены неисправного предохранителя соответствующий бит не сбрасывается автоматически. Его необходимо стереть путем сброса сообщения об ошибке.</li> </ul> 	С (ошибка)	новый	Q00, Q00 и Q01
SD131						
SD132						
SD133						
SD134						
SD135						
SD136						
SD137	Модули ввода-вывода с ошибкой сверки	Набор битов (16 бит), показывает модули с ошибкой сверки 0: ошибок сверки модулей ввода-вывода нет 1: имеется ошибка сверки модуля ввода-вывода	<ul style="list-style-type: none"> <li>Если текущее состояние модуля ввода-вывода отличается от заданного состояния после включения напряжения питания, то информация этого модуля ввода-вывода сохраняется в регистре. (Если в параметрах установлен номер модуля, то сохраняется этот номер.)</li> <li>Распознается также информация модуля ввода-вывода.</li> </ul> 	С (ошибка)	новый	
SD150						
SD151						
SD152						
SD153						
SD154						
SD155						
SD156						
SD157						



(2) Системная информация

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:				
SD200	Коммутационное состояние	Состояние переключателя режимов на модуле центрального процессора	<ul style="list-style-type: none"> <li>Состояние переключателя режимов сохраняется в следующем формате:</li> </ul> <table border="1"> <tr> <td>(1) режим</td> <td>всегда 1: STOP</td> </tr> </table>	(1) режим	всегда 1: STOP	С (постоянно)	новый	Rem		
			(1) режим	всегда 1: STOP						
			<ul style="list-style-type: none"> <li>Состояние переключателя режимов сохраняется в следующем формате:</li> </ul> <table border="1"> <tr> <td>(1) режим</td> <td>(0): RUN (1): STOP</td> </tr> <tr> <td>(2) выключатель карты памяти</td> <td>всегда ВЫКЛ.</td> </tr> </table>	(1) режим	(0): RUN (1): STOP	(2) выключатель карты памяти	всегда ВЫКЛ.		новый	Q00, Q00 и Q01
			(1) режим	(0): RUN (1): STOP						
(2) выключатель карты памяти	всегда ВЫКЛ.									
<ul style="list-style-type: none"> <li>Состояние переключателя режимов сохраняется в следующем формате:</li> </ul> <table border="1"> <tr> <td>(1) режим</td> <td>(0): RUN (1): STOP (2): L.CLR</td> </tr> <tr> <td>(2) выключатель карты памяти</td> <td>всегда ВЫКЛ.</td> </tr> <tr> <td>(3) микро-выключатели</td> <td>Биты с b8 по b11 соответствуют выключателям с SW1 по SW5. 0: ВЫКЛ., 1: ВКЛ.</td> </tr> </table>	(1) режим	(0): RUN (1): STOP (2): L.CLR	(2) выключатель карты памяти	всегда ВЫКЛ.	(3) микро-выключатели	Биты с b8 по b11 соответствуют выключателям с SW1 по SW5. 0: ВЫКЛ., 1: ВКЛ.	С (обработка команды END)	новый	Q кроме Q00, Q00 и Q01	
(1) режим	(0): RUN (1): STOP (2): L.CLR									
(2) выключатель карты памяти	всегда ВЫКЛ.									
(3) микро-выключатели	Биты с b8 по b11 соответствуют выключателям с SW1 по SW5. 0: ВЫКЛ., 1: ВКЛ.									
<ul style="list-style-type: none"> <li>Состояние выключателя с ключом на модуле центрального процессора сохраняется в следующем формате:</li> </ul> <table border="1"> <tr> <td>(1) Режим</td> <td>(0): RUN (1): STOP (2): L.CLR</td> </tr> <tr> <td>(2) Выключатели карты памяти</td> <td>b4: карта памяти "А" b5: карта памяти "В" 0: выключ., 1: включено</td> </tr> <tr> <td>(3) Микровыключатели</td> <td>Биты от b8 до b11 соответствуют выключателям от SW1 до SW5. 0: ВЫКЛ., 1: ВКЛ.</td> </tr> </table>	(1) Режим	(0): RUN (1): STOP (2): L.CLR	(2) Выключатели карты памяти	b4: карта памяти "А" b5: карта памяти "В" 0: выключ., 1: включено	(3) Микровыключатели	Биты от b8 до b11 соответствуют выключателям от SW1 до SW5. 0: ВЫКЛ., 1: ВКЛ.	С (обработка команды END)	новый	QnA	
(1) Режим	(0): RUN (1): STOP (2): L.CLR									
(2) Выключатели карты памяти	b4: карта памяти "А" b5: карта памяти "В" 0: выключ., 1: включено									
(3) Микровыключатели	Биты от b8 до b11 соответствуют выключателям от SW1 до SW5. 0: ВЫКЛ., 1: ВКЛ.									

(2) Системная информация

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:
SD201	Состояние светодиода	Состояние светодиодного индикатора на модуле центрального процессора	<ul style="list-style-type: none"> <li>Нижеприведенная информация относится к светодиодным индикаторам центрального процессора.</li> </ul> <p>(1) : RUN (5) : BOOT                  (2) : ERROR (6) : свободно                  (3) : USER (7) : свободно                  (4) : BAT.ALARM (8) : режим</p> <p>Режим сохраняется в виде следующего набора битов:                  0: не горит,                  1: зеленый,                  2: оранжевый</p> <p>В процессорах Q00J, Q00 и Q01 имеются только области 1 и 2.</p>	С (изменение состояния)	новый	Q
			<ul style="list-style-type: none"> <li>Нижеприведенная информация относится к светодиодным индикаторам центрального процессора и сохраняется в виде следующего набора битов:</li> <li>выкл. при 0; вкл. при 1; мигание при 2</li> </ul> <p>(1) : RUN (5) : BOOT                  (2) : ERROR (6) : Card A (карта памяти "А")                  (3) : USER (7) : Card B (карта памяти "В")                  (4) : BAT.ALARM (8) : свободно</p>	С (изменение состояния)	новый	QnA
SD202	Выключенные светодиоды	Набор битов выключенных светодиодов	<ul style="list-style-type: none"> <li>Сохраняет набор битов выключенных светодиодов (возможно только в отношении светодиодов "USER" и "BOOT")</li> <li>выключен при 0, включен при 1</li> </ul>	П	новый	QnA
SD203	Состояние обработки центрального процессора		<ul style="list-style-type: none"> <li>Состояние обработки сохраняется в следующем формате:</li> </ul> <p>(1) состояние обработки децентрализованных модулей ввода-вывода всегда 2: STOP</p>	С (постоянно)	новый	Rem
			<ul style="list-style-type: none"> <li>Состояние обработки центрального процессора сохраняется в следующем формате.</li> </ul> <p>(1) : Сост. обраб. центр. проц.                  0 : RUN                  1 : STEP-RUN (кроме Q00J, Q00 и Q01)                  2 : STOP                  3 : PAUSE</p> <p>(2) : Режим STOP/PAUSE вызван                  0 : переключателем режимов                  1 : дистанционным контактом                  2 : периферийным устройством, соединением с компьютером или другим удаленным источником                  3 : внутренними прогр. командами                  4 : ошибкой</p> <p>Примечание: сохраняется только ошибка, возникшая первой.</p>	С (обработка команды END)	D9015 (изменился формат)	●

(2) Системная информация

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:																								
SD206	Вид тестирования операндов	Отображение выполненного теста операндов	<ul style="list-style-type: none"> <li>При тестировании операндов с помощью программатора этот регистр содержит информацию о том, какие операнды были протестированы.</li> <li>0: тестирование операндов не активировано</li> <li>1: тестирование входов (X)</li> <li>2: тестирование выходов (Y)</li> <li>3: тестирование входов и выходов (X/Y)</li> </ul>	С (запрос)	новый	Rem																								
SD207	Приоритет индикации светодиода "ERR"	Приоритет от 1 до 4	<ul style="list-style-type: none"> <li>При возникновении ошибки она отображается миганием светодиодного индикатора в соответствии с номером ошибки, сохраненным в регистрах.</li> <li>Для приоритетов индикации возможны следующие настройки:</li> </ul> <table border="1" style="margin-left: 40px;"> <tr> <td></td> <td>B15</td> <td>B12 B11</td> <td>B8 B7</td> <td>B4 B3</td> <td>B0</td> </tr> <tr> <td>SD207</td> <td>приор. 4</td> <td>приор. 3</td> <td>приор. 2</td> <td>приор. 1</td> <td></td> </tr> <tr> <td>SD208</td> <td>приор. 8</td> <td>приор. 7</td> <td>приоритет 6</td> <td>приор. 5</td> <td></td> </tr> <tr> <td>SD209</td> <td></td> <td></td> <td>приор. 10</td> <td>приор. 9</td> <td></td> </tr> </table> <p style="margin-left: 40px;">Заводская настройка: (4321h) (8765 h) (00A9 h)</p> <ul style="list-style-type: none"> <li>Если настройка равна „0“, индикация не происходит. Но даже при настройке "0" светодиод отображает информацию об ошибке, вызвавшую останов центрального процессора (отображаются также настройки параметров).</li> </ul>		B15	B12 B11	B8 B7	B4 B3	B0	SD207	приор. 4	приор. 3	приор. 2	приор. 1		SD208	приор. 8	приор. 7	приоритет 6	приор. 5		SD209			приор. 10	приор. 9		П	D9038	<ul style="list-style-type: none"> <li>кроме Q00, Q00 и Q01</li> </ul>
		B15		B12 B11	B8 B7	B4 B3	B0																							
SD207		приор. 4		приор. 3	приор. 2	приор. 1																								
SD208	приор. 8	приор. 7	приоритет 6	приор. 5																										
SD209			приор. 10	приор. 9																										
SD208	Приоритет от 5 до 8	D9039 (изменился формат)																												
SD209	Приоритет от 9 до 10	новый																												
SD210	Данные времени	Данные времени (год, месяц)	<ul style="list-style-type: none"> <li>Год (последние 2 разряда) и месяц сохраняются в двоично-десятичной кодировке в регистре SD210:</li> </ul> <table border="1" style="margin-left: 40px;"> <tr> <td>b15</td><td>b12 b11</td><td>b8 b7</td><td>b4 b3</td><td>b0</td> </tr> <tr> <td colspan="2">год</td><td colspan="3">месяц</td> </tr> </table> <p style="margin-left: 40px;">Пример: июль 1993 г. = 9307</p>	b15	b12 b11	b8 b7	b4 b3	b0	год		месяц			С/П (запрос)	D9025	<ul style="list-style-type: none"> <li>Rem</li> </ul>														
b15		b12 b11	b8 b7	b4 b3	b0																									
год		месяц																												
SD211	Данные времени (день, час)	<ul style="list-style-type: none"> <li>День и час сохраняются в двоично-десятичной кодировке в регистре SD211:</li> </ul> <table border="1" style="margin-left: 40px;"> <tr> <td>b15</td><td>b12 b11</td><td>b8 b7</td><td>b4 b3</td><td>b0</td> </tr> <tr> <td colspan="2">день</td><td colspan="3">час</td> </tr> </table> <p style="margin-left: 40px;">Пример: 31-е число, 10 часов = 3110</p>	b15	b12 b11	b8 b7	b4 b3	b0	день		час			D9026																	
b15	b12 b11	b8 b7	b4 b3	b0																										
день		час																												
SD212	Данные времени (минута, секунда)	<ul style="list-style-type: none"> <li>Минуты и секунды сохраняются в двоично-десятичной кодировке в регистре SD212:</li> </ul> <table border="1" style="margin-left: 40px;"> <tr> <td>b15</td><td>b12 b11</td><td>b8 b7</td><td>b4 b3</td><td>b0</td> </tr> <tr> <td colspan="2">минута</td><td colspan="3">секунда</td> </tr> </table> <p style="margin-left: 40px;">Пример: 35 минут, 48 секунд = 3548</p>	b15	b12 b11	b8 b7	b4 b3	b0	минута		секунда			D9027																	
b15	b12 b11	b8 b7	b4 b3	b0																										
минута		секунда																												

(2) Системная информация

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:		
SD213	Данные времени	Данные времени (день недели)	<ul style="list-style-type: none"> <li>● День недели сохраняется в двоично-десятичной кодировке в регистре SD213.</li> </ul>	С/П (запрос)	D9028	Q Rem		
			<ul style="list-style-type: none"> <li>● День недели сохраняется в двоично-десятичной кодировке в регистре SD213.</li> </ul>	С/П (запрос)		QnA		
SD220	Данные светодиодного дисплея	Данные индикации на дисплее	<ul style="list-style-type: none"> <li>● Данные в формате ASCII (16 знаков) светодиодного дисплея сохраняются в нижеуказанных регистрах.</li> </ul>	С (изменение состояния)	новый	●		
SD221			SD220				15-й знак справа	16-й знак справа
SD222			SD221				13-й знак справа	14-й знак справа
SD223			SD222				11-й знак справа	12-й знак справа
SD224			SD223				9-й знак справа	10-й знак справа
SD226			SD224				7-й знак справа	8-й знак справа
SD227			SD225				5-й знак справа	6-й знак справа
			SD226				3-й знак справа	4-й знак справа
SD227	SD227	1-й знак справа	2-й знак справа					
SD240	Режим монтажной шины	0: автоматический режим 1: детализированный режим	Этот регистр служит для сохранения режима монтажной шины.	С (инициализация)	новый	Q Rem		
SD241	Количество расширительных монтажных шин	0: только главная монтажная шина от 1 до 7: количество расширит. монтаж. шин	В этом регистре сохраняется количество установленных расширительных монтажных шин.	С (инициализация)	новый			

(2) Системная информация

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:										
SD242	Различение монтажных шин "А" и "Q"	0: Установлена шина типа QA [ ] [ ] B (режим А) 1: Установлена шина типа Q [ ] [ ] B (режим Q)	<p>Если расширительная монтажная шина не подсоединена, биты с 1-го по 4-й имеют состояние "0".</p>	С (инициализация)	новый	Q00J, Q00 и Q01										
			<p>Если расширительная монтажная шина не подсоединена, биты с 1-го по 7-й имеют состояние "0".</p>			Q02, Q06H, Q12H, Q25H; Rem										
SD243	Количество слотов на монтажных шинах	Количество слотов на монтажных шинах. В случае процессоров Q00J, Q00 и Q01 разряды с 5-го по 7-й расширительной шины заняты нулями.	<table border="1"> <tr> <td>bF</td> <td>bC bB</td> <td>b8 b7</td> <td>b4 b3</td> <td>b0</td> </tr> <tr> <td>3-я РМШ</td> <td>2-я РМШ</td> <td>1-я РМШ</td> <td>ГМШ</td> <td></td> </tr> </table>	bF	bC bB	b8 b7	b4 b3	b0	3-я РМШ	2-я РМШ	1-я РМШ	ГМШ		С (инициализация)	новый	Q
bF			bC bB	b8 b7	b4 b3	b0										
3-я РМШ	2-я РМШ	1-я РМШ	ГМШ													
SM244	<table border="1"> <tr> <td>7-я РМШ</td> <td>6-я РМШ</td> <td>5-я РМШ</td> <td>4-я РМШ</td> </tr> </table> <p>Количество слотов укладывается указывается для главной монтажной шины (ГМШ) и расширительных монтажных шин (РМШ) в соответствующих областях.</p>	7-я РМШ	6-я РМШ	5-я РМШ	4-я РМШ											
7-я РМШ	6-я РМШ	5-я РМШ	4-я РМШ													
SD250	Загружено максимальное число входов и выходов	Загружено максимальное количество входов и выходов	Если SM250 установлен, к двум старшим разрядам последнего загруженного адреса модуля ввода-вывода прибавляется 1 и результат сохраняется в виде двоичного значения.	С (обработка команды END)	новый	●										
SD251	Адрес заменяемого модуля ввода-вывода	Начальный адрес модуля ввода-вывода	Регистр D9094 сохраняет в виде двоичного значения два старших разряда начального адреса модуля ввода-вывода, который во время режима онлайн извлекается из монтажной шины или вставляется в нее.	П	D9094	Q2A(S1), Q3A, Q4A, Q4AR										
SD253	Скорость передачи для RS422	0: 9600 бит/с 1: 19.2 кбит/с 2: 38.4 кбит/с	Этот регистр сохраняет скорость передачи для интерфейса RS422.	С (при изменении)	новый	QnA										

(2) Системная информация

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:			
SD254	Информация MELSECNET/10	Количество установленных модулей	Показывает количество модулей, установленных в сети MELSECNET/10.	С (инициализация)	новый	●			
SD255		Информация первого модуля	адрес ввода-вывода				Адрес ввода-вывода первого модуля, установленного в сети MELSECNET/10.		
SD256			номер сети				Сетевой адрес первого модуля, установленного в сети MELSECNET/10.		
SD257			номер группы				Номер группы первого модуля, установленного в сети MELSECNET/10.		
SD258			номер станции				Номер станции первого модуля, установленного в сети MELSECNET/10.		
SD259			информация о резервировании				Если имеются резервные станции, сохраняется номер модуля резервной станции (от 1 до 4).		
SD260 – SD264		Информация второго модуля	Конфигурация идентична первому модулю.						● кроме Q001, Q00 и Q01
SD265 – SD269		Информация третьего модуля							
SD270 – SD274		Информация четвертого модуля							
SD280	Ошибка CC-Link	Состояние при обнаружении ошибки	<p>(1) При включении Xn0 установленной станции CC-Link, устанавливается сопоставленный этой станции бит.</p> <p>(2) Если выключается Xn1 или XnF установленной станции CC-Link, устанавливается сопоставленный этой станции бит.</p> <p>(3) Биты в этой области устанавливаются, если центральный процессор не может коммуницировать с установленной станцией CC-Link.</p>	С (при ошибке)	новый	Q Rem			
			<p>(1) Если включается Xn0 установленной станции CC-Link, устанавливается сопоставленный этой станции бит.</p> <p>(2) Если выключается Xn1 или XnF установленной сети CC-Link, устанавливается сопоставленный этой станции бит.</p> <p>(3) Биты в этой области устанавливаются, если центральный процессор не может коммуницировать с установленной станцией CC-Link.</p>				С (при ошибке)	новый	QnA

## (2) Системная информация

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:	
SD290	Присвоение операндов (идентично содержимому параметров)	Число адресов операнда X	Настроенное в данный момент число адресов операндов X	С (инициализация)	новый	● Rem	
SD291		Число адресов операндов Y	Настроенное в данный момент число адресов операндов Y				
SD292		Число адресов операндов M	Настроенное в данный момент число адресов операндов M				
SD293		Число адресов операндов L	Настроенное в данный момент число адресов операндов L		новый	●	
SD294		Число адресов операндов B	Настроенное в данный момент число адресов операндов B		новый	● Rem	
SD295		Число адресов операндов F	Настроенное в данный момент число адресов операндов F		новый	●	
SD296		Число адресов операндов SB	Настроенное в данный момент число адресов операндов SB		новый	● Rem	
SD297		Число адресов операндов V	Настроенное в данный момент число адресов операндов V		новый	●	
SD298		Число адресов операндов S	Настроенное в данный момент число адресов операндов S				
SD299		Число адресов операндов T	Настроенное в данный момент число адресов операндов T				
SD300		Число адресов операндов ST	Настроенное в данный момент число адресов операндов ST				
SD301		Число адресов операндов C	Настроенное в данный момент число адресов операндов C		(инициализация)	новый	● Rem
SD302		Число адресов операндов D	Настроенное в данный момент число адресов операндов D				
SD303		Число адресов операндов W	Настроенное в данный момент число адресов операндов W				
SD304		Число адресов операндов SW	Настроенное в данный момент число адресов операндов SW				
SD315	Время, зарезервированное для коммуникации	Время, зарезервированное для коммуникации.	Введенное здесь время (диапазон от 1 до 100 мс) выделено для коммуникации с программатором. Чем большее значение здесь введено, тем короче время, имеющееся для реакции при коммуникации с другими приборами (например, через последовательное соединение). Если это значение находится вне допустимого диапазона, оно игнорируется. Время цикла удлиняется на настроенное время.	Обработка команды END	новый	Q	

(2) Системная информация

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:	
SD340	Информация Ethernet	Количество установленных модулей	Показывает количество модулей, установленных в Ethernet.	С (инициализация)	новый	Q Rem	
SD341		Информация первого модуля	адрес ввода-вывода				Адрес ввода-вывода первого модуля, установленного в сети Ethernet.
SD342			номер сети				Сетевой адрес первого модуля, установленного в сети Ethernet.
SD343			номер группы				Номер группы первого модуля, установленного в сети Ethernet.
SD344			номер станции				Номер станции первого модуля, установленного в сети Ethernet.
SD345 – SD346			свободно		Не используется. В случае процессора серии "Q" IP-адрес первого модуля Ethernet сохраняется в буферной памяти.		
SD347		свободно	Не используется. В случае процессора серии "Q" код ошибки Ethernet первого модуля считывается с помощью команды ERRORRD.				
SD348 – SD354		Информация второго модуля	Конфигурация идентична первому модулю.		новый	Q (кроме Q00J, Q00 и Q01) Rem	
SD355 – SD361		Информация третьего модуля	Конфигурация идентична первому модулю.				
SD362 – SD368		Информация четвертого модуля	Конфигурация идентична первому модулю.				
SD340		Информация Ethernet	Количество установленных модулей		Показывает количество модулей, установленных в Ethernet.	С (инициализация)	новый
SD341	Информация первого модуля		адрес ввода-вывода	Адрес ввода-вывода первого модуля, установленного в сети Ethernet.			
SD342			номер сети	Номер сети первого модуля, установленного в сети Ethernet.			
SD343			номер группы	Номер группы первого модуля, установленного в сети Ethernet.			
SD344			номер станции	Номер станции первого модуля, установленного в сети Ethernet.			
SD345 – SD346			IP-адрес	IP-адрес модуля Ethernet, установленного первым.			
SD347	код ошибки		Код ошибки Ethernet модуля, установленного первым.				
SD348 – SD354	Информация второго модуля		Конфигурация идентична первому модулю.				
SD355 – SD361	Информация третьего модуля						
SD362 – SD368	Информация четвертого модуля						



(2) Системная информация

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:
SD380	Состояние приема команд Ethernet	Информация первого модуля	<p>Бит установлен: принято (канал используется) Бит не установлен: не принято (канал не используется)</p>	С (инициализация)	новый	Q
SD381		Информация второго модуля	Конфигурация идентична первому модулю.			
SD382		Информация третьего модуля				
SD383		Информация четвертого модуля				
SD392	Версия программного обеспечения	Версия системного программного обеспечения	Версия внутреннего системного программного обеспечения сохраняется в старшем байте в кодировке ASCII. Данные младшего байта недействительны. Например, программное обеспечение версии "А" в старшем байте сохраняется в виде кода 41 <sub>н</sub> . Сохраненный здесь номер версии системного программного обеспечения может отличаться от номера, напечатанного на корпусе модуля.	С (инициализация)	D9060	
SD395	Номер процессора	1: центральный процессор 1 2: центральный процессор 2 3: центральный процессор 3 4: центральный процессор 4	Этот регистр содержит номер центрального процессора, если он используется в многопроцессорной системе.	С (инициализация)	новый	центр. проц. серии "System Q", пригодный для многопроцессорного режима

## (3) Системные часы и счетчики

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:
SD412	1-секундный счетчик	Считает с секундным шагом	<ul style="list-style-type: none"> <li>● При начале режима "RUN" центрального процессора счетчик начинает считать с секундным тактом.</li> <li>● Счетчик считает вверх от 0 до 32767, затем перескакивает на -32768 и считает до 0.</li> </ul>	С (изменение состояния)	D9022	●
SD414	Такт в 2п секунд	Единицы по 2п секунд	<ul style="list-style-type: none"> <li>● Сохраняет настройку для п, на основе которой рассчитывается такт в 2п секунд (предварительная настройка = 30).</li> <li>● Возможны значения между 1 и 32767.</li> </ul>	П	новый	
SD415	Такт в 2п миллисекунд	Единицы по 2п миллисекунд	<ul style="list-style-type: none"> <li>● Сохраняет настройку для п, на основе которой вычисляется такт в 2п миллисекунд (предварительная настройка = 30).</li> <li>● Возможны значения между 1 и 32767.</li> </ul>	П	новый	Q02, Q02H, Q06H, Q12PH, Q12PH, Q25H, Q25PH
SD420	Счетчик программных циклов	Считает количество программных циклов	<ul style="list-style-type: none"> <li>● После начала режима "RUN" центрального процессора счетчик при каждом программном цикле повышается на 1.</li> <li>● Счетчик считает вверх от 0 до 32767, затем перескакивает на -32768 и считает до 0.</li> </ul>	С (обработка команды END)	новый	●
SD430	Счетчик программных циклов замедленной обработки	Считает количество программных циклов, выполненных на пониженной скорости	<ul style="list-style-type: none"> <li>● После включения центрального процессора в режиме RUN счетчик при каждом программном цикле прирастает на 1.</li> <li>● Счетчик считает вверх от 0 до 32767, затем перескакивает на -32768 и считает до 0.</li> <li>● Этот счетчик можно использовать только для программ, выполняемых в замедленном режиме („Low Speed Execution“).</li> </ul>	С (обработка команды END)	новый	● кроме Q00J, Q00 и Q01

(4) Информация цикла программы

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:
SD500	Номер выполняемой программы	Тип выполнения программы, выполняемой в данный момент	<ul style="list-style-type: none"> <li>● Номер выполняемой в данный момент программы сохраняется в виде двоичного значения.</li> </ul>	С (изменение состояния)	новый	● кроме Q00, Q00 и Q01
SD510	Номер программы "Low Speed Execution"	Название файла программы	<ul style="list-style-type: none"> <li>● Номер программы, выполняемой в данный момент в замедленном режиме (Low Speed Execution), сохранен в виде двоичного значения.</li> <li>● Это возможно только при установленном маркере SM510.</li> </ul>	С (обработка команды END)	новый	
SD520	Текущее время цикла	Время цикла (единица 1 мс)	<ul style="list-style-type: none"> <li>● Сохраняет текущее время цикла программы (с шагом в 1 мс) в диапазоне от 0 до 65535.</li> <li>● Сохраняет текущее время цикла программы (с шагом в 1 мкс) в диапазоне от 00000 до 900.</li> </ul> Пример: время цикла программы 23.6 мс сохраняется следующим образом: D520 = 23 D521 = 600	С (обработка команды END)	D9017 (изменился формат)	●
SD521		Время цикла (единица 1 мкс)			новый	
SD522	Время цикла инициализации	Время цикла инициализации (единица 1 мс)	<ul style="list-style-type: none"> <li>● Сохраняет время первого программного цикла (с шагом в 1 мс).</li> <li>● Диапазон от 0 до 65535</li> </ul>	С (обработка команды END)	новый	● кроме Q00, Q00 и Q01
SD523		Время цикла инициализации (единица 100 мкс)				
SD524	Минимальное время цикла	Минимальное время цикла (единица 1 мс)	<ul style="list-style-type: none"> <li>● Сохраняет минимальное время цикла программы (с шагом в 1 мс).</li> <li>● Диапазон от 0 до 65535.</li> </ul>	С (обработка команды END)	D9018 (изменился формат)	●
SD525		Минимальное время цикла (единица 100 мкс)			<ul style="list-style-type: none"> <li>● Сохраняет минимальное время цикла программы (с шагом в 100 мкс).</li> <li>● Диапазон от 000 до 900.</li> </ul>	
SD526	Максимальное время цикла	Максимальное время цикла (единица 1 мс)	<ul style="list-style-type: none"> <li>● Сохраняет максимальное время цикла программы (с шагом в 1 мс), за исключением первого цикла.</li> <li>● Диапазон от 0 до 65535</li> </ul>	С (обработка команды END)	D9019 (изменился формат)	●
SD527		Максимальное время цикла (единица 100 мкс)			<ul style="list-style-type: none"> <li>● Сохраняет максимальное время цикла программы (с шагом в 100 мкс), за исключением первого цикла.</li> <li>● Диапазон от 000 до 900</li> </ul>	
SD528	Время цикла для программ, выполняемых в режиме "Low Speed Execution"	Текущее время цикла (единица 1 мс)	<ul style="list-style-type: none"> <li>● Сохраняет текущее время цикла программы, выполняемой в замедленном режиме "Low Speed Execution" (с шагом в 1 мс).</li> <li>● Диапазон от 0 до 65535.</li> </ul>	С (обработка команды END)	новый	
SD529		Текущее время цикла (единица 100 мкс)				
SD532	Минимальное время цикла для программ, выполняемых в режиме "Low Speed Execution"	Минимальное время цикла (единица 1 мс)	<ul style="list-style-type: none"> <li>● Сохраняет минимальное время цикла программы, выполняемой в замедленном режиме "Low Speed Execution" (с шагом в 1 мс).</li> <li>● Диапазон от 0 до 65535.</li> </ul>	С (обработка команды END)	новый	● кроме Q00, Q00 и Q01
SD533		Минимальное время цикла (единица 100 мкс)				
SD534	Максимальное время цикла для программ, выполняемых в режиме "Low Speed Execution"	Максимальное время цикла (единица 1 мс)	<ul style="list-style-type: none"> <li>● Сохраняет максимальное время цикла программы, выполняемой в замедленном режиме "Low Speed Execution" (с шагом в 1 мс, за исключением 1-го цикла).</li> <li>● Диапазон от 0 до 65535.</li> </ul>	С (обработка команды END)	новый	
SD535		Максимальное время цикла (единица 100 мкс)				

## (4) Информация цикла программы

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:
SD540	Время обработки команды END	Время обработки команды END (единица 1 мс)	<ul style="list-style-type: none"> <li>Сохраняет время от конца последнего программного цикла до начала следующего цикла (с шагом в 1 мс).</li> <li>Диапазон от 0 до 65535.</li> </ul>	C (обработка команды END)	новый	
SD541		Время обработки команды END (единица 100 мкс)	<ul style="list-style-type: none"> <li>Сохраняет время от конца последнего программного цикла до начала следующего цикла (с шагом в 100 мкс).</li> <li>Диапазон от 000 до 900.</li> </ul>			
SD542	Время ожидания при постоянном времени цикла	Время ожидания при постоянном времени цикла (единица 1 мс)	<ul style="list-style-type: none"> <li>Сохраняет время ожидания при установленном постоянном времени цикла (с шагом в 1 мс).</li> <li>Диапазон от 0 до 65535.</li> </ul>	C (первая END)	новый	●
SD543		Время ожидания при постоянном времени цикла (единица 100 мкс)	<ul style="list-style-type: none"> <li>Сохраняет время ожидания при установленном постоянном времени цикла (с шагом в 100 мкс).</li> <li>Диапазон от 000 до 900.</li> </ul>			
SD544	Суммарное время выполнения программ в режиме "Low Speed Execution"	Суммарное время выполнения программ, выполняемых в замедленном режиме (единица 1 мс)	<ul style="list-style-type: none"> <li>Сохраняет суммарное время выполнения программы, выполняемой в замедленном режиме "Low Speed Execution" (с шагом в 1 мс).</li> <li>Диапазон от 0 до 65535.</li> </ul>	C (обработка команды END)	новый	●
SD545		Суммарное время выполнения программ, выполняемых в режиме "Low Speed Execution" (единица 100 мкс)	<ul style="list-style-type: none"> <li>Сохраняет суммарное время выполнения программы, выполняемой в замедленном режиме "Low Speed Execution" (с шагом в 100 мкс).</li> <li>Диапазон от 000 до 900.</li> </ul>			
SD546	Время выполнения программ, выполняемых в режиме "Low Speed Execution"	Время выполнения программ, выполняемых в режиме "Low Speed Execution" (единица 1 мс)	<ul style="list-style-type: none"> <li>Сохраняет время выполнения программы, выполняемой в замедленном режиме "Low Speed Execution" (с шагом в 1 мс), на протяжении одного цикла.</li> <li>Диапазон от 0 до 65535.</li> <li>Сохранение происходит каждый цикл.</li> </ul>	C (обработка команды END)	новый	● кроме Q00J, Q00 и Q01
SD547		Время выполнения для программ, выполняемых в режиме "Low Speed Execution" (единица 100 мкс)	<ul style="list-style-type: none"> <li>Сохраняет время выполнения программы, выполняемой в замедленном режиме "Low Speed Execution" (с шагом в 100 мкс), на протяжении одного цикла.</li> <li>Диапазон от 000 до 900.</li> <li>Сохранение происходит каждый цикл.</li> </ul>			
SD548	Время выполнения программ, выполняемых в режиме "Scan Execution"	Время выполнения для программ, выполняемых в режиме "Scan Execution" (единица 1 мс)	<ul style="list-style-type: none"> <li>Сохраняет время выполнения программы, выполняемой в режиме "Scan Execution" (с шагом в 1 мс), на протяжении одного цикла.</li> <li>Диапазон от 0 до 65535.</li> <li>Сохранение происходит каждый цикл.</li> </ul>	C (обработка команды END)	новый	●
SD549		Время выполнения для программ, выполняемых в режиме "Scan Execution" (единица 100 мкс)	<ul style="list-style-type: none"> <li>Сохраняет время выполнения программы, выполняемой в режиме "Scan Execution" (с шагом в 100 мкс), на протяжении одного цикла.</li> <li>Диапазон от 000 до 900.</li> <li>Сохранение происходит каждый цикл.</li> </ul>			
SD550	Измерение интервала интервала сервисного обслуживания для модулей	№ станции/модуля	<ul style="list-style-type: none"> <li>Устанавливает адрес ввода-вывода модуля, для которого измеряется интервал сервисного обслуживания.</li> </ul>	П	новый	
SD551	Интервал сервисного обслуживания	Интервал сервисного обслуживания модуля (единица 1 мс)	<ul style="list-style-type: none"> <li>Если маркер SM551 установлен, то сохраняется интервал, по истечении которого должно быть выполнено обслуживание модуля, указанного в SD550 (с шагом в 1 мс).</li> <li>Диапазон от 0 до 65535.</li> </ul>	C (запрос)	новый	● кроме Q00J, Q00 и Q01
SD552		Интервал сервисного обслуживания модуля (единица 100 мкс)	<ul style="list-style-type: none"> <li>Если SM551 установлен, то сохраняется интервал, по истечении которого должно быть выполнено обслуживание модуля, указанного в SD550 (с шагом в 100 мкс).</li> <li>Диапазон от 000 до 900.</li> </ul>			

(5) Карты памяти

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:																
SD600	Тип карты памяти "А"		<p>Показывает тип установленной карты памяти "А".</p>	С (при инициализации и извлечении карты памяти)	новый	серия "Q", кроме Q00, Q00 и Q01																
			<p>Показывает тип установленной карты памяти "А".</p>	С (при инициализации и извлечении карты памяти)	новый	QnA																
SD602	Емкость дисковода 1 (RAM)		Емкость дисковода 1 сохраняется с шагом в 1 кб.	С (при инициализации и извлечении карты памяти)	новый	● кроме Q00, Q00 и Q01																
SD603	Емкость дисковода 2 (ROM)		Емкость дисковода 2 сохраняется с шагом в 1 кб.	С (при инициализации и извлечении карты памяти)	новый	● кроме Q00, Q00 и Q01																
SD604	Условия использования карты памяти "А"		<ul style="list-style-type: none"> <li>Условия использования карты памяти "А" сохраняются в виде набора битов ("вкл." соответствует использованию).</li> <li>Значение этого набора битов:</li> </ul> <table border="1"> <tr> <td>b0 : операция первонач. загрузки (QBT)</td> <td>b8 : протокол ошибок центр. процессора (QFD)</td> </tr> <tr> <td>b1 : параметры (QPT)</td> <td>b9 : трассировка SFC (QTS)</td> </tr> <tr> <td>b2 : комментарии к операндам (QCD)</td> <td>bA : локальная переменная (QDL)</td> </tr> <tr> <td>b3 : начальное значение операнда (QDI)</td> <td>bC :</td> </tr> <tr> <td>b4 : регистр файлов (QDR)</td> <td>bD :</td> </tr> <tr> <td>b5 : трассировка (QTS)</td> <td>bE :</td> </tr> <tr> <td>b6 :</td> <td>bF :</td> </tr> <tr> <td>b7 :</td> <td></td> </tr> </table>	b0 : операция первонач. загрузки (QBT)	b8 : протокол ошибок центр. процессора (QFD)	b1 : параметры (QPT)	b9 : трассировка SFC (QTS)	b2 : комментарии к операндам (QCD)	bA : локальная переменная (QDL)	b3 : начальное значение операнда (QDI)	bC :	b4 : регистр файлов (QDR)	bD :	b5 : трассировка (QTS)	bE :	b6 :	bF :	b7 :		С (изменение состояния)	новый	серия "Q", кроме Q00, Q00 и Q01
			b0 : операция первонач. загрузки (QBT)	b8 : протокол ошибок центр. процессора (QFD)																		
b1 : параметры (QPT)	b9 : трассировка SFC (QTS)																					
b2 : комментарии к операндам (QCD)	bA : локальная переменная (QDL)																					
b3 : начальное значение операнда (QDI)	bC :																					
b4 : регистр файлов (QDR)	bD :																					
b5 : трассировка (QTS)	bE :																					
b6 :	bF :																					
b7 :																						
<ul style="list-style-type: none"> <li>Условия использования карты памяти "А" сохраняются в виде набора битов ("вкл." соответствует использованию).</li> <li>Значение этого набора битов:</li> </ul> <table border="1"> <tr> <td>b0 : процесс первонач. загр. (QBT)</td> <td>b8 : данные имитации</td> </tr> <tr> <td>b1 : параметры (QPA)</td> <td>b9 : протокол ошибок ЦП (QFD)</td> </tr> <tr> <td>b2 : комментарии к опер. (QCD)</td> <td>b10 : трассир. языка SFC (QTS)</td> </tr> <tr> <td>b3 : нач. значение операнда (QDI)</td> <td>b11 : лок. переменная (QDL)</td> </tr> <tr> <td>b4 : регистр файлов (QDR)</td> <td>b12 :</td> </tr> <tr> <td>b5 : трассировка (QTS)</td> <td>b13 :</td> </tr> <tr> <td>b6 : фиксация состояния (QTL)</td> <td>b14 :</td> </tr> <tr> <td>b7 : трассировка программы (QTP)</td> <td>b15 :</td> </tr> </table>	b0 : процесс первонач. загр. (QBT)	b8 : данные имитации	b1 : параметры (QPA)	b9 : протокол ошибок ЦП (QFD)	b2 : комментарии к опер. (QCD)	b10 : трассир. языка SFC (QTS)	b3 : нач. значение операнда (QDI)	b11 : лок. переменная (QDL)	b4 : регистр файлов (QDR)	b12 :	b5 : трассировка (QTS)	b13 :	b6 : фиксация состояния (QTL)	b14 :	b7 : трассировка программы (QTP)	b15 :	С (изменение состояния)	новый	QnA			
b0 : процесс первонач. загр. (QBT)	b8 : данные имитации																					
b1 : параметры (QPA)	b9 : протокол ошибок ЦП (QFD)																					
b2 : комментарии к опер. (QCD)	b10 : трассир. языка SFC (QTS)																					
b3 : нач. значение операнда (QDI)	b11 : лок. переменная (QDL)																					
b4 : регистр файлов (QDR)	b12 :																					
b5 : трассировка (QTS)	b13 :																					
b6 : фиксация состояния (QTL)	b14 :																					
b7 : трассировка программы (QTP)	b15 :																					

(5) Карты памяти

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:																
SD620	Тип карты памяти "В"		<ul style="list-style-type: none"> <li>Показывает тип установленной карты памяти "В".</li> </ul> <p>В связи с тем, что в процессор встроена флэш-РОМ, значение для дисквода 4 неизменно установлено на "3".</p> <table border="1"> <tr> <td>дисквод 1 (RAM)</td> <td>0: не имеется 1: SRAM</td> </tr> <tr> <td>дисквод 2 (ROM)</td> <td>0: не имеется (1: SRAM) 2: ATA FLASH 3: FLASH ROM</td> </tr> </table>	дисквод 1 (RAM)	0: не имеется 1: SRAM	дисквод 2 (ROM)	0: не имеется (1: SRAM) 2: ATA FLASH 3: FLASH ROM	С (инициализация)	новый	Q												
			дисквод 1 (RAM)	0: не имеется 1: SRAM																		
дисквод 2 (ROM)	0: не имеется (1: SRAM) 2: ATA FLASH 3: FLASH ROM																					
			<ul style="list-style-type: none"> <li>Показывает тип установленной карты памяти "В".</li> </ul> <table border="1"> <tr> <td>дисквод 1 (RAM)</td> <td>0: не имеется 1: SRAM</td> </tr> <tr> <td>дисквод 2 (ROM)</td> <td>0: не имеется 2: EEPROM 3: FLASH ROM</td> </tr> </table>	дисквод 1 (RAM)	0: не имеется 1: SRAM	дисквод 2 (ROM)	0: не имеется 2: EEPROM 3: FLASH ROM	С (инициализация)	новый	Q2A(S1), Q3A, Q4A, Q4AR												
дисквод 1 (RAM)	0: не имеется 1: SRAM																					
дисквод 2 (ROM)	0: не имеется 2: EEPROM 3: FLASH ROM																					
SD622	Емкость дисквода 3 (RAM)		<ul style="list-style-type: none"> <li>Свободное место на дискводе 3 регистрируется с шагом в 1 кб. В случае процессора серии "Q" с RAM емкостью 61 кб это значение неизменно установлено на "61".</li> </ul>	С (инициализация)	новый	Q																
			<ul style="list-style-type: none"> <li>Свободное место на дискводе 3 регистрируется с шагом в 1 кб.</li> </ul>	С (инициализация)	новый	Q2A(S1), Q3A, Q4A, Q4AR																
SD623	Емкость дисквода 4 (ROM)		<ul style="list-style-type: none"> <li>Емкость дисквода 4 сохраняется с шагом в 1 кб.</li> </ul>	С (инициализация)	новый	серия "Q", Q2A(S1), Q3A, Q4A, Q4AR																
SD624	Условия использования дисквода 3		<ul style="list-style-type: none"> <li>Условия использования дисквода 3 отображаются с помощью бита 4: бит 4 = выкл.: не используется бит 4 = вкл.: используется для хранения регистров файлов</li> </ul>	С (изменение состояния)	новый	Q00J, Q00 и Q01																
	Условия использования дискводов 3 и 4		<ul style="list-style-type: none"> <li>Условия использования дискводов 3 и 4 сохраняются в виде набора битов ("вкл." соответствует использованию).</li> <li>Значение этого набора битов разъяснено ниже.</li> </ul> <table border="1"> <tr> <td>b0 : операция первонач. загрузки (QBT)</td> <td>b8 : параметры (QPA)</td> </tr> <tr> <td>b1 : параметры (QPA)</td> <td>b9 : протокол ошибок центр. процессора (QFD)</td> </tr> <tr> <td>b2 : комментарии к операндам (QCD)</td> <td>bA : трассировка SFC (QTS)</td> </tr> <tr> <td>b3 : начальное значение операнда (QDI)</td> <td>bB : локальная переменная (QDL)</td> </tr> <tr> <td>b4 : регистр файлов (QDR)</td> <td>bC :</td> </tr> <tr> <td>b5 : трассировка (QTS)</td> <td>bD :</td> </tr> <tr> <td>b6 :</td> <td>bE :</td> </tr> <tr> <td>b7 :</td> <td>bF :</td> </tr> </table>	b0 : операция первонач. загрузки (QBT)	b8 : параметры (QPA)	b1 : параметры (QPA)	b9 : протокол ошибок центр. процессора (QFD)	b2 : комментарии к операндам (QCD)	bA : трассировка SFC (QTS)	b3 : начальное значение операнда (QDI)	bB : локальная переменная (QDL)	b4 : регистр файлов (QDR)	bC :	b5 : трассировка (QTS)	bD :	b6 :	bE :	b7 :	bF :	С (изменение состояния)	новый	серия "Q", кроме Q00J, Q00 и Q01
	b0 : операция первонач. загрузки (QBT)	b8 : параметры (QPA)																				
b1 : параметры (QPA)	b9 : протокол ошибок центр. процессора (QFD)																					
b2 : комментарии к операндам (QCD)	bA : трассировка SFC (QTS)																					
b3 : начальное значение операнда (QDI)	bB : локальная переменная (QDL)																					
b4 : регистр файлов (QDR)	bC :																					
b5 : трассировка (QTS)	bD :																					
b6 :	bE :																					
b7 :	bF :																					
Условия использования карты памяти "В"		<ul style="list-style-type: none"> <li>Условия использования карты памяти "В" сохраняются в виде набора битов ("вкл." соответствует использованию).</li> <li>Значение этого набора битов разъяснено ниже.</li> </ul> <table border="1"> <tr> <td>b0 : процесс первонач. загр. (QBT)</td> <td>b8 : данные имитации</td> </tr> <tr> <td>b1 : параметры (QPA)</td> <td>b9 : протокол ошибок ЦП (QFD)</td> </tr> <tr> <td>b2 : комментарии к опер. (QCD)</td> <td>b10 : трассир. языка SFC (QTS)</td> </tr> <tr> <td>b3 : нач. значение операнда (QDI)</td> <td>b11 : лок. переменная (QDL)</td> </tr> <tr> <td>b4 : регистр файлов (QDR)</td> <td>b12 :</td> </tr> <tr> <td>b5 : трассировка (QTS)</td> <td>b13 :</td> </tr> <tr> <td>b6 : фиксация состояния (QTL)</td> <td>b14 :</td> </tr> <tr> <td>b7 : трассировка программы (QTP)</td> <td>b15 :</td> </tr> </table>	b0 : процесс первонач. загр. (QBT)	b8 : данные имитации	b1 : параметры (QPA)	b9 : протокол ошибок ЦП (QFD)	b2 : комментарии к опер. (QCD)	b10 : трассир. языка SFC (QTS)	b3 : нач. значение операнда (QDI)	b11 : лок. переменная (QDL)	b4 : регистр файлов (QDR)	b12 :	b5 : трассировка (QTS)	b13 :	b6 : фиксация состояния (QTL)	b14 :	b7 : трассировка программы (QTP)	b15 :	С (изменение состояния)	новый	Q2A(S1), Q3A, Q4A, Q4AR	
b0 : процесс первонач. загр. (QBT)	b8 : данные имитации																					
b1 : параметры (QPA)	b9 : протокол ошибок ЦП (QFD)																					
b2 : комментарии к опер. (QCD)	b10 : трассир. языка SFC (QTS)																					
b3 : нач. значение операнда (QDI)	b11 : лок. переменная (QDL)																					
b4 : регистр файлов (QDR)	b12 :																					
b5 : трассировка (QTS)	b13 :																					
b6 : фиксация состояния (QTL)	b14 :																					
b7 : трассировка программы (QTP)	b15 :																					

(5) Карты памяти

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:																						
SD640	Дисковод регистра файлов	Номер дисковода	Сохраняет номер дисковода, используемого для регистра файлов.	C (изменение состояния)	новый																							
SD641	Регистр файлов Название файла		Сохраняет регистр файлов и название файла (с расширением) в ASCII-кодировке, указанные с помощью параметра или команды QCDSET.  <table border="1"> <tr> <td>b15</td> <td>b8 b7</td> <td>b0</td> </tr> <tr> <td>SD641</td> <td>2-й знак</td> <td>1-й знак</td> </tr> <tr> <td>SD642</td> <td>4-й знак</td> <td>3-й знак</td> </tr> <tr> <td>SD643</td> <td>6-й знак</td> <td>5-й знак</td> </tr> <tr> <td>SD644</td> <td>8-й знак</td> <td>7-й знак</td> </tr> <tr> <td>SD645</td> <td>1-й знак расширения</td> <td>2EH (.)</td> </tr> <tr> <td>SD646</td> <td>3-й знак расширения</td> <td>2-й знак расширения</td> </tr> </table>	b15	b8 b7	b0	SD641	2-й знак	1-й знак	SD642	4-й знак	3-й знак	SD643	6-й знак	5-й знак	SD644	8-й знак	7-й знак	SD645	1-й знак расширения	2EH (.)	SD646	3-й знак расширения	2-й знак расширения	C (изменение состояния)	новый	●	
b15				b8 b7	b0																							
SD641				2-й знак	1-й знак																							
SD642				4-й знак	3-й знак																							
SD643				6-й знак	5-й знак																							
SD644				8-й знак	7-й знак																							
SD645	1-й знак расширения	2EH (.)																										
SD646	3-й знак расширения	2-й знак расширения																										
SD642																												
SD643																												
SD644																												
SD645																												
SD646																												
SD647	Объем регистра файлов		Объем выбранного в данный момент регистра файлов в килобайтах.	C (изменение состояния)	новый																							
SD648	Номер блока регистра файлов		Сохраняет выбранный в данный момент номер блока регистра файлов.	C (изменение состояния)	D9035																							
SD650	Дисковод комментариев		Сохраняет номер дисковода комментариев, указанный с помощью параметра или команды QCDSET.	C (изменение состояния)	новый																							
SD651	Название файла комментариев		Сохраняет название файла (с расширением) в кодировке ASCII, указанное с помощью параметра или команды QCDSET.  <table border="1"> <tr> <td>b15</td> <td>b8 b7</td> <td>b0</td> </tr> <tr> <td>SD651</td> <td>2-й знак</td> <td>1-й знак</td> </tr> <tr> <td>SD652</td> <td>4-й знак</td> <td>3-й знак</td> </tr> <tr> <td>SD653</td> <td>6-й знак</td> <td>5-й знак</td> </tr> <tr> <td>SD654</td> <td>8-й знак</td> <td>7-й знак</td> </tr> <tr> <td>SD655</td> <td>1-й знак расширения</td> <td>2EH (.)</td> </tr> <tr> <td>SD656</td> <td>3-й знак расширения</td> <td>2-й знак расширения</td> </tr> </table>	b15	b8 b7	b0	SD651	2-й знак	1-й знак	SD652	4-й знак	3-й знак	SD653	6-й знак	5-й знак	SD654	8-й знак	7-й знак	SD655	1-й знак расширения	2EH (.)	SD656	3-й знак расширения	2-й знак расширения	C (изменение состояния)	новый	●	
b15				b8 b7	b0																							
SD651				2-й знак	1-й знак																							
SD652				4-й знак	3-й знак																							
SD653				6-й знак	5-й знак																							
SD654				8-й знак	7-й знак																							
SD655	1-й знак расширения	2EH (.)																										
SD656	3-й знак расширения	2-й знак расширения																										
SD652																												
SD653																												
SD654																												
SD655																												
SD656																												
SD660	Файл, указанный для процесса начальной загрузки	Номер дисковода, в котором находится указанный файл для начальной загрузки	Сохраняет номер дисковода, в котором находится файл (*.QBT), указанный для процесса загрузки.	C (инициализация)	новый	● кроме Q00, Q00 и Q01																						
SD661		Название файла, указанного для процесса начальной загрузки		Сохраняет название файла, указанного для процесса начальной загрузки (*.QBT).  <table border="1"> <tr> <td>b15</td> <td>b8 b7</td> <td>b0</td> </tr> <tr> <td>SD661</td> <td>2-й знак</td> <td>1-й знак</td> </tr> <tr> <td>SD662</td> <td>4-й знак</td> <td>3-й знак</td> </tr> <tr> <td>SD663</td> <td>6-й знак</td> <td>5-й знак</td> </tr> <tr> <td>SD664</td> <td>8-й знак</td> <td>7-й знак</td> </tr> <tr> <td>SD665</td> <td>1-й знак расширения</td> <td>2EH (.)</td> </tr> <tr> <td>SD666</td> <td>3-й знак расширения</td> <td>2-й знак расширения</td> </tr> </table>	b15		b8 b7	b0	SD661	2-й знак	1-й знак	SD662	4-й знак	3-й знак	SD663	6-й знак	5-й знак	SD664	8-й знак	7-й знак	SD665	1-й знак расширения	2EH (.)	SD666	3-й знак расширения	2-й знак расширения	C (инициализация)	новый
b15					b8 b7		b0																					
SD661					2-й знак		1-й знак																					
SD662					4-й знак		3-й знак																					
SD663					6-й знак		5-й знак																					
SD664	8-й знак	7-й знак																										
SD665	1-й знак расширения	2EH (.)																										
SD666	3-й знак расширения	2-й знак расширения																										
SD662																												
SD663																												
SD664																												
SD665																												
SD666																												

(6) Регистры, относящиеся к командам

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:
SD705	Битовая схема		Во время обработки блока устанавливается маркер SM705. Это позволяет использовать сохраненную в регистре SD705 битовую схему (в случае применения двойных слов она сохранена в регистрах SD705 и SD706) для всех обрабатываемых данных блока.	П	новый	● кроме Q00, Q00 и Q01
SD706						
SD714	Количество свободных запросов коммуникации в области регистрации	0 ... 32	Количество свободных блоков в области коммуникационного запроса для удаленных специальных модулей, соединенных с AJ71PT32-5.	С (во время выполнения)	M9081	QnA
SD715	Битовая схема команды IMASK	Битовая схема	При применении команды IMASK используется следующая битовая схема.  b15 ..... b0 SD715  15; 14; 13; 12; 11; 10;  9;  8;  7;  6;  5;  4;  3;  2;  1;  0 SD716  31; 30; 29; 28; 27; 26; 25; 24; 23; 22; 21; 20; 19; 18; 17; 16 SD717  47; 46; 45; 44; 43; 42; 41; 40; 39; 38; 37; 36; 35; 34; 33; 32	С (во время выполнения)	новый	●
SD716						
SD717						
SD718	Сумматор		Эти регистры эмулируют сумматоры контроллеров MELSEC серии "А".	С/П	новый	
SD719						
SD720	Присвоение номера программы для команды PLOAD		Этот регистр сохраняет номер программы, который должен быть присвоен программе, загружаемой с помощью команды PLOAD. Возможны номера программ от 1 до 124.	П	новый	Q
SD730	Количество свободных запросов коммуникации CC-Link в области регистрации	0 ... 32	Сохраняет количество свободных блоков в области коммуникационного запроса CC-Link для удаленных специальных модулей, соединенных с A(1S)J61QBT61.	С (во время выполнения)	новый	QnA
SD736	Ввод с помощью PKEY		Этот регистр диагностики временно сохраняет данные, введенные с клавиатуры с помощью команды PKEY.	С (во время выполнения)	новый	● кроме Q00, Q00 и Q01



(6) Регистры, относящиеся к командам

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:																																																																																																
SD738 SD739 SD740 SD741 SD742 SD743 SD744 SD745 SD746 SD747 SD748 SD749 SD750 SD751 SD752 SD753 SD754 SD755 SD756 SD757 SD758 SD759 SD760 SD761 SD762 SD763 SD764 SD765 SD766 SD767 SD768 SD769	Память сообщений		<p>Сохраняет сообщения, сформулированные с помощью команды MSG.</p> <p style="text-align: center;">b15 ←──────────→ b8 b7 ←──────────→ b0</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr><td>SD738</td><td>2-й знак</td><td>1-й знак</td></tr> <tr><td>SD739</td><td>4-й знак</td><td>3-й знак</td></tr> <tr><td>SD740</td><td>6-й знак</td><td>5-й знак</td></tr> <tr><td>SD741</td><td>8-й знак</td><td>7-й знак</td></tr> <tr><td>SD742</td><td>10-й знак</td><td>9-й знак</td></tr> <tr><td>SD743</td><td>12-й знак</td><td>11-й знак</td></tr> <tr><td>SD744</td><td>14-й знак</td><td>13-й знак</td></tr> <tr><td>SD745</td><td>16-й знак</td><td>15-й знак</td></tr> <tr><td>SD746</td><td>18-й знак</td><td>17-й знак</td></tr> <tr><td>SD747</td><td>20-й знак</td><td>19-й знак</td></tr> <tr><td>SD748</td><td>22-й знак</td><td>21-й знак</td></tr> <tr><td>SD749</td><td>24-й знак</td><td>23-й знак</td></tr> <tr><td>SD750</td><td>26-й знак</td><td>25-й знак</td></tr> <tr><td>SD751</td><td>28-й знак</td><td>27-й знак</td></tr> <tr><td>SD752</td><td>30-й знак</td><td>29-й знак</td></tr> <tr><td>SD753</td><td>32-й знак</td><td>31-й знак</td></tr> <tr><td>SD754</td><td>34-й знак</td><td>33-й знак</td></tr> <tr><td>SD755</td><td>36-й знак</td><td>35-й знак</td></tr> <tr><td>SD756</td><td>38-й знак</td><td>37-й знак</td></tr> <tr><td>SD757</td><td>40-й знак</td><td>39-й знак</td></tr> <tr><td>SD758</td><td>42-й знак</td><td>41-й знак</td></tr> <tr><td>SD759</td><td>44-й знак</td><td>43-й знак</td></tr> <tr><td>SD760</td><td>46-й знак</td><td>45-й знак</td></tr> <tr><td>SD761</td><td>48-й знак</td><td>47-й знак</td></tr> <tr><td>SD762</td><td>50-й знак</td><td>49-й знак</td></tr> <tr><td>SD763</td><td>52-й знак</td><td>51-й знак</td></tr> <tr><td>SD764</td><td>54-й знак</td><td>53-й знак</td></tr> <tr><td>SD765</td><td>56-й знак</td><td>55-й знак</td></tr> <tr><td>SD766</td><td>58-й знак</td><td>57-й знак</td></tr> <tr><td>SD767</td><td>60-й знак</td><td>59-й знак</td></tr> <tr><td>SD768</td><td>62-й знак</td><td>61-й знак</td></tr> <tr><td>SD769</td><td>64-й знак</td><td>63-й знак</td></tr> </table>	SD738	2-й знак	1-й знак	SD739	4-й знак	3-й знак	SD740	6-й знак	5-й знак	SD741	8-й знак	7-й знак	SD742	10-й знак	9-й знак	SD743	12-й знак	11-й знак	SD744	14-й знак	13-й знак	SD745	16-й знак	15-й знак	SD746	18-й знак	17-й знак	SD747	20-й знак	19-й знак	SD748	22-й знак	21-й знак	SD749	24-й знак	23-й знак	SD750	26-й знак	25-й знак	SD751	28-й знак	27-й знак	SD752	30-й знак	29-й знак	SD753	32-й знак	31-й знак	SD754	34-й знак	33-й знак	SD755	36-й знак	35-й знак	SD756	38-й знак	37-й знак	SD757	40-й знак	39-й знак	SD758	42-й знак	41-й знак	SD759	44-й знак	43-й знак	SD760	46-й знак	45-й знак	SD761	48-й знак	47-й знак	SD762	50-й знак	49-й знак	SD763	52-й знак	51-й знак	SD764	54-й знак	53-й знак	SD765	56-й знак	55-й знак	SD766	58-й знак	57-й знак	SD767	60-й знак	59-й знак	SD768	62-й знак	61-й знак	SD769	64-й знак	63-й знак	С (во время выполнения)	новый	● кроме Q00J, Q00 и Q01
SD738			2-й знак	1-й знак																																																																																																		
SD739			4-й знак	3-й знак																																																																																																		
SD740			6-й знак	5-й знак																																																																																																		
SD741			8-й знак	7-й знак																																																																																																		
SD742			10-й знак	9-й знак																																																																																																		
SD743			12-й знак	11-й знак																																																																																																		
SD744			14-й знак	13-й знак																																																																																																		
SD745			16-й знак	15-й знак																																																																																																		
SD746			18-й знак	17-й знак																																																																																																		
SD747			20-й знак	19-й знак																																																																																																		
SD748			22-й знак	21-й знак																																																																																																		
SD749			24-й знак	23-й знак																																																																																																		
SD750			26-й знак	25-й знак																																																																																																		
SD751			28-й знак	27-й знак																																																																																																		
SD752			30-й знак	29-й знак																																																																																																		
SD753			32-й знак	31-й знак																																																																																																		
SD754			34-й знак	33-й знак																																																																																																		
SD755			36-й знак	35-й знак																																																																																																		
SD756			38-й знак	37-й знак																																																																																																		
SD757			40-й знак	39-й знак																																																																																																		
SD758			42-й знак	41-й знак																																																																																																		
SD759			44-й знак	43-й знак																																																																																																		
SD760			46-й знак	45-й знак																																																																																																		
SD761			48-й знак	47-й знак																																																																																																		
SD762			50-й знак	49-й знак																																																																																																		
SD763			52-й знак	51-й знак																																																																																																		
SD764			54-й знак	53-й знак																																																																																																		
SD765			56-й знак	55-й знак																																																																																																		
SD766			58-й знак	57-й знак																																																																																																		
SD767			60-й знак	59-й знак																																																																																																		
SD768			62-й знак	61-й знак																																																																																																		
SD769	64-й знак	63-й знак																																																																																																				
SD774 – SD775	Пределы для ПИД-регуляторов	<p>0: пределы для ПИД-регуляторов установлены</p> <p>1: пределы не установлены</p>	<p>Пределы сопоставлены отдельным контурам ПИД-регулирования следующим образом:</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">b15</td> <td></td> <td style="text-align: center;">b1</td> <td style="text-align: center;">b0</td> </tr> <tr> <td>SD774</td> <td>контур рег.16</td> <td>...</td> <td>контур рег. 2    контур рег. 1</td> </tr> <tr> <td>SD775</td> <td>контур рег.32</td> <td>...</td> <td>контур рег.18    контур рег.17</td> </tr> </table>	b15		b1	b0	SD774	контур рег.16	...	контур рег. 2    контур рег. 1	SD775	контур рег.32	...	контур рег.18    контур рег.17	П	новый	серия "Q", кроме Q00J, Q00 и Q01																																																																																				
b15		b1	b0																																																																																																			
SD774	контур рег.16	...	контур рег. 2    контур рег. 1																																																																																																			
SD775	контур рег.32	...	контур рег.18    контур рег.17																																																																																																			
SD780	Остающееся количество команд, относящихся к CC-Link, которые могут выполняться одновременно	0 ... 32	В этом регистре сохраняется остающееся количество команд, относящихся к CC-Link, которые могут выполняться одновременно.	П	новый	QnA																																																																																																
SD781 – SD793	Битовая схема команды IMASK	Битовая схема	<p>При применении команды IMASK используется следующая битовая схема.</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">b15</td> <td></td> <td style="text-align: center;">b11</td> <td style="text-align: center;">b0</td> </tr> <tr> <td>SD781</td> <td>I63</td> <td>до</td> <td>I59    I48</td> </tr> <tr> <td>SD782</td> <td>I79</td> <td>до</td> <td>I65    I64</td> </tr> <tr> <td colspan="4" style="text-align: center;">до</td> </tr> <tr> <td>SD793</td> <td>I255</td> <td>до</td> <td>I241    I240</td> </tr> </table> <p>В процессорах Q00J, Q00 и Q01 можно использовать только область от SD781 (I48) до SD785 (I127).</p>	b15		b11	b0	SD781	I63	до	I59    I48	SD782	I79	до	I65    I64	до				SD793	I255	до	I241    I240	С (во время выполнения)	новый	Q																																																																												
b15		b11	b0																																																																																																			
SD781	I63	до	I59    I48																																																																																																			
SD782	I79	до	I65    I64																																																																																																			
до																																																																																																						
SD793	I255	до	I241    I240																																																																																																			

(7) Устранение ошибок

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [[]][[]]	Дейст. для:
SD806	Название файла для фиксации состояния		Название файла (с расширением) сохраняется в кодировке ASCII в момент фиксации состояния. b15                                  b8 b7                                  b0 SD806    2-й знак                                  1-й знак SD807    4-й знак                                  3-й знак SD808    6-й знак                                  5-й знак SD809    8-й знак                                  7-й знак SD810    1-й знак расширения                  2ЕН (.) SD811    3-й знак расширения                  2-й знак расширения	С (во время выполнения)	новый	
SD807						
SD808						
SD809						
SD810						
SD811						
SD812						
SD813						
SD814						
SD815						
SD816						
SD816	* Конфигурация: 					

(8) Область фиксации

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ]	Дейст. для:																																																			
SD900	Номер дисковода во время исчезновения напряжения	Номер дисковода с файлом, доступ к которому осуществлялся в момент исчезновения напряжения	Сохраняет номер дисковода с файлом, к которому осуществлялся доступ во время исчезновения напряжения.	С (изменение состояния)	новый	QnA																																																			
SD901	Название активных файлов в момент исчезновения напряжения	Название файла, к которому осуществлялся доступ в момент исчезновения напряжения	Сохраняет название файла в кодировке ASCII (с расширением), к которому осуществлялся доступ в момент исчезновения напряжения.  <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%; text-align: center;">b15</td> <td style="width: 33%; text-align: center;">b8 b7</td> <td style="width: 33%; text-align: center;">b0</td> </tr> <tr> <td>SD901</td> <td style="text-align: center;">2-й знак</td> <td style="text-align: center;">1-й знак</td> </tr> <tr> <td>SD902</td> <td style="text-align: center;">4-й знак</td> <td style="text-align: center;">3-й знак</td> </tr> <tr> <td>SD903</td> <td style="text-align: center;">6-й знак</td> <td style="text-align: center;">5-й знак</td> </tr> <tr> <td>SD904</td> <td style="text-align: center;">8-й знак</td> <td style="text-align: center;">7-й знак</td> </tr> <tr> <td>SD905</td> <td style="text-align: center;">1-й знак расширения</td> <td style="text-align: center;">2ЕН (.)</td> </tr> <tr> <td>SD906</td> <td style="text-align: center;">3-й знак расширения</td> <td style="text-align: center;">2-й знак расширения</td> </tr> </table>	b15	b8 b7		b0	SD901	2-й знак	1-й знак	SD902	4-й знак	3-й знак	SD903	6-й знак	5-й знак	SD904	8-й знак	7-й знак	SD905	1-й знак расширения	2ЕН (.)	SD906	3-й знак расширения	2-й знак расширения	С (изменение состояния)	новый																														
b15				b8 b7	b0																																																				
SD901				2-й знак	1-й знак																																																				
SD902				4-й знак	3-й знак																																																				
SD903				6-й знак	5-й знак																																																				
SD904				8-й знак	7-й знак																																																				
SD905	1-й знак расширения	2ЕН (.)																																																							
SD906	3-й знак расширения	2-й знак расширения																																																							
SD902																																																									
SD903																																																									
SD904																																																									
SD905																																																									
SD906																																																									
SD910	Ввод с помощью PKEY		Последовательно сохраняет коды, введенные с пульта PU.  <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%; text-align: center;">b15</td> <td style="width: 33%; text-align: center;">b8 b7</td> <td style="width: 33%; text-align: center;">b0</td> </tr> <tr> <td>SD910</td> <td style="text-align: center;">2-й знак</td> <td style="text-align: center;">1-й знак</td> </tr> <tr> <td>SD911</td> <td style="text-align: center;">4-й знак</td> <td style="text-align: center;">3-й знак</td> </tr> <tr> <td>SD912</td> <td style="text-align: center;">6-й знак</td> <td style="text-align: center;">5-й знак</td> </tr> <tr> <td>SD913</td> <td style="text-align: center;">8-й знак</td> <td style="text-align: center;">7-й знак</td> </tr> <tr> <td>SD914</td> <td style="text-align: center;">10-й знак</td> <td style="text-align: center;">9-й знак</td> </tr> <tr> <td>SD915</td> <td style="text-align: center;">12-й знак</td> <td style="text-align: center;">11-й знак</td> </tr> <tr> <td>SD916</td> <td style="text-align: center;">14-й знак</td> <td style="text-align: center;">13-й знак</td> </tr> <tr> <td>SD917</td> <td style="text-align: center;">16-й знак</td> <td style="text-align: center;">15-й знак</td> </tr> <tr> <td>SD918</td> <td style="text-align: center;">18-й знак</td> <td style="text-align: center;">17-й знак</td> </tr> <tr> <td>SD919</td> <td style="text-align: center;">20-й знак</td> <td style="text-align: center;">19-й знак</td> </tr> <tr> <td>SD920</td> <td style="text-align: center;">22-й знак</td> <td style="text-align: center;">21-й знак</td> </tr> <tr> <td>SD921</td> <td style="text-align: center;">24-й знак</td> <td style="text-align: center;">23-й знак</td> </tr> <tr> <td>SD922</td> <td style="text-align: center;">26-й знак</td> <td style="text-align: center;">25-й знак</td> </tr> <tr> <td>SD923</td> <td style="text-align: center;">28-й знак</td> <td style="text-align: center;">27-й знак</td> </tr> <tr> <td>SD924</td> <td style="text-align: center;">30-й знак</td> <td style="text-align: center;">29-й знак</td> </tr> <tr> <td>SD925</td> <td style="text-align: center;">32-й знак</td> <td style="text-align: center;">31-й знак</td> </tr> </table>	b15	b8 b7		b0	SD910	2-й знак	1-й знак	SD911	4-й знак	3-й знак	SD912	6-й знак	5-й знак	SD913	8-й знак	7-й знак	SD914	10-й знак	9-й знак	SD915	12-й знак	11-й знак	SD916	14-й знак	13-й знак	SD917	16-й знак	15-й знак	SD918	18-й знак	17-й знак	SD919	20-й знак	19-й знак	SD920	22-й знак	21-й знак	SD921	24-й знак	23-й знак	SD922	26-й знак	25-й знак	SD923	28-й знак	27-й знак	SD924	30-й знак	29-й знак	SD925	32-й знак	31-й знак	С (Во время выполнения)	новый
b15				b8 b7	b0																																																				
SD910				2-й знак	1-й знак																																																				
SD911				4-й знак	3-й знак																																																				
SD912				6-й знак	5-й знак																																																				
SD913				8-й знак	7-й знак																																																				
SD914				10-й знак	9-й знак																																																				
SD915				12-й знак	11-й знак																																																				
SD916				14-й знак	13-й знак																																																				
SD917				16-й знак	15-й знак																																																				
SD918				18-й знак	17-й знак																																																				
SD919				20-й знак	19-й знак																																																				
SD920				22-й знак	21-й знак																																																				
SD921				24-й знак	23-й знак																																																				
SD922				26-й знак	25-й знак																																																				
SD923				28-й знак	27-й знак																																																				
SD924	30-й знак	29-й знак																																																							
SD925	32-й знак	31-й знак																																																							
SD911																																																									
SD912																																																									
SD913																																																									
SD914																																																									
SD915																																																									
SD916																																																									
SD917																																																									
SD918																																																									
SD919																																																									
SD920																																																									
SD921																																																									
SD922																																																									
SD923																																																									
SD924																																																									
SD925																																																									

(9) Модули с неисправным предохранителем или отсутствующим внешним электропитанием

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [ ] [ ] [ ] [ ]	Действ. для:																																																																
SD1300	Модули с неисправным предохранителем	Набор битов (16 бит) показывает модули с неисправным предохранителем 0: Неисправных предохранителей нет 1: Имеется неисправный предохранитель	<ul style="list-style-type: none"> <li>Количество выходных модулей с неисправным предохранителем сохраняется в виде набора 16 битов. (Если номер модуля установлен в параметрах, сохраняется этот номер.)</li> <li>Неисправные предохранители распознаются и в модулях вывода удаленных станций.</li> <li>После замены неисправного предохранителя соответствующий бит не сбрасывается автоматически. Его необходимо стереть путем сброса сообщения об ошибке.</li> </ul> <div style="text-align: center;"> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>Sd1301</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table> <p style="margin-left: 40px;">↓</p> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>Sd1331</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> </table> <p style="margin-left: 40px;">↑ индикация неисправного предохранителя</p> </div>	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	Sd1301	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	Sd1331	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	С (ошибка)	D9100	● кроме Q00, Q00 и Q01
b15				b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																				
0				0	0	1	0	0	0	0	0	0	0	0	0	0	0	0																																																				
Sd1301				1	0	0	0	0	0	1	0	0	0	0	0	0	0	0																																																				
Sd1331				0	1	0	0	0	0	0	0	0	0	0	0	1	0	0																																																				
SD1301				D9101																																																																		
SD1302				D9102																																																																		
SD1303				D9103																																																																		
SD1304				D9104																																																																		
SD1305				D9105																																																																		
SD1306				D9106																																																																		
SD1307	D9107																																																																					
SD1308	новый																																																																					
SD1309 – SD1330	новый																																																																					
SD1331	новый																																																																					
SD1350 – SD1381	Отсутствует внешнее электропитание	Набор битов (16 бит) показывает модули, в которых отсутствует внешнее электропитание. 0: Внешнее электропитание отсутствует 1: Внешнее электропитание имеется	<p>Адреса выходных модулей, в которых отсутствует внешнее электропитание, сохраняются в виде набора битов (в единицах по 16 битов). (Если номер модуля настроен в параметрах, сохраняется этот номер.)</p> <div style="text-align: center;"> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>SD1351</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table> <p style="margin-left: 40px;">↓</p> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr><td>SD1381</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> </table> <p style="margin-left: 40px;">↑ В модуле с адресом Y1F80 нет внешнего электропитания.</p> </div>	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	SD1351	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	SD1381	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	С (ошибка)	новый	серия "Q", кроме Q00, Q00 и Q01
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																							
0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0																																																							
SD1351	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0																																																							
SD1381	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0																																																							

(10) Сверка модулей ввода-вывода

Адрес	Название	Значение	Описание	Устанавливает (если установлен)	Регистр проц. серии "А" D9 [1][1]	Дейст. для:																																																																																					
SD1400	Модули ввода-вывода с ошибкой сверки	Набор битов (16 бит), показывает модули с ошибкой сверки 0 : модуль ввода-вывода не имеет ошибки сверки 1 : модуль ввода-вывода имеет ошибку сверки	<ul style="list-style-type: none"> <li>● Если фактическое состояние модуля ввода-вывода отличается от заданного состояния после включения напряжения питания, информация этого модуля ввода-вывода сохраняется в регистре. (Если номер модуля настроен в параметрах, сохраняется этот номер.)</li> <li>● Распознается также информация модуля ввода-вывода.</li> </ul> <div style="text-align: center;"> <table border="1" style="margin: auto;"> <tr> <td></td> <td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>SD1400</td> <td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> <tr> <td>SD1401</td> <td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> <tr> <td>↓</td> <td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td> </tr> <tr> <td>SD1431</td> <td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p style="text-align: center;">↑ индикация режима ввода-вывода с ошибкой сравнения</p> </div>		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	SD1400	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	SD1401	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	↓	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	SD1431	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	С (ошибка)	D9116	● кроме Q001, Q00 и Q01
				b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																								
SD1400				0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0																																																																								
SD1401				1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0																																																																								
↓				:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:																																																																								
SD1431				0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0																																																																								
SD1401				D9117																																																																																							
SD1402				D9118																																																																																							
SD1403				D9119																																																																																							
SD1404				D9120																																																																																							
SD1405				D9121																																																																																							
SD1406	D9122																																																																																										
SD1407	D9123																																																																																										
SD1408	новый																																																																																										
SD1409 до SD1430	новый																																																																																										
SD1431	новый																																																																																										

(11) Взаимосвязь между специальными регистрами (серия "А") и регистрами диагностики (серия "Q"/"System Q")

При переходе с контроллеров MELSEC серии "А" на контроллеры MELSEC серии "Q" специальные регистры D9000...D9255 серии "А" соответствуют регистрам диагностики SD1000...SD1255 серий "Q" и "System Q".

Все эти регистры диагностики устанавливаются системой. Они не могут быть изменены с помощью пользовательской программы. Если пользователь желает устанавливать или сбрасывать эти регистры, он должен изменить свои программы так, чтобы в них использовались только настоящие регистры диагностики Q/QnA.

Исключением являются специальные регистры D9200...D9255. Так как пользователь имеет возможность записи в эти регистры, запись в соответствующие конвертированные регистры диагностики SD1200...SD1255 также возможна.

Подробную информацию о специальных регистрах серии "А" можно найти в руководствах по центральным процессорам и сети MELSECNET.

**ПРИМЕЧАНИЕ** Если в таблице указан эквивалентный регистр диагностики для центральных процессоров серий Q и QnA, то программу следует изменить и использовать этот регистр. Если эквивалентный регистр диагностики Q/QnA не указан, можно использовать конвертированный регистр.

Перечень специальных регистров и регистров диагностики

Специальный регистр процессора серии "А"	Конвертированный регистр диагностики	Эквивалентный регистр диагностики "System Q"/QnA	Название	Значение	Дейст. для:
D9000	SD1000	—	Неисправен предохранитель	Адрес модуля с неисправным предохранителем	Q и QnA
D9001	SD1001	—	Неисправен предохранитель	Адрес модуля с неисправным предохранителем	
D9002	SD1002	—	Ошибка сверки модуля ввода-вывода	Адрес модуля, в отношении которого имеется ошибка сверки	
D9004	SD1004	—	Ошибка в мастер-модуле сети MELSECNET/MINI S3	Сохраняет состояние, установленное в параметрах (модули от 1 до 8)	QnA
D9005	SD1005	—	Количество исчезновений сетевого напряжения	Счетчик исчезновений напряжения	Q и QnA
D9008	SD1008	SD0	Распознавание ошибок путем самодиагностики	Номер ошибки, выявленной путем самодиагностики	
D9009	SD1009	SD62	Распознавание маркера ошибки	Маркер ошибки F для обозначения внешней ошибки	
D9010	SD1010	в процессорах "System Q" и QnA не действует	Номер шага программы, содержащего ошибку	Номер шага, в котором возникла ошибка	
D9011	SD1011				
D9014	SD1014				
D9015	SD1015	SD203	Рабочее состояние центрального процессора	Рабочее состояние центрального процессора	
D9016	SD1016	в процессорах "System Q" и QnA не действует	Обозначение программы	Обозначение основной программы, обрабатываемой в данный момент	
D9017	SD1017	SD520	Время цикла	Минимальное время цикла (единица "10 мс")	

## Перечень специальных регистров и регистров диагностики

Специальный регистр процессора серии "А"	Конвертированный регистр диагностики	Эквивалентный регистр диагностики "System Q"/QnA	Название	Значение	Дейст. для:
D9018	SD1018	SD524	Время цикла	Время цикла (единица "10 мс")	Q и QnA
D9019	SD1019	SD526	Время цикла	Максимальное время цикла (единица "10 мс")	
D9020	SD1020	в процессорах "System Q" и QnA не действует	Постоянное время цикла	Заданное значение постоянного времени цикла (пользовательская настройка с шагом в 10 мс)	
D9021	SD1021	—	Время цикла	Время цикла (единица "1 мс")	
D9022	SD1022	SD412	1-секундный счетчик	1-секундный счетчик	
D9025	SD1025	SD210	Данные часов	Данные часов (год, месяц)	
D9026	SD1026	SD211		Данные часов (день, час)	
D9027	SD1027	SD212		Данные часов (минута, секунда)	
D9028	SD1028	SD213		Данные часов (день недели)	
D9035	SD1035	SD648	Расширенный регистр файлов	Номер используемого блока	
D9036	SD1036	в процессорах "System Q" и QnA не действует	Адреса операндов расширенных регистров файлов	Адреса операндов расширенных регистров файлов для непосредственного считывания и записи	
D9037	SD1037				
D9038	SD1038	SD207	Приоритет индикации на светодиодном дисплее	Приоритет от 1 до 4	
D9039	SD1039	SD208		Приоритет от 5 до 7	
D9044	SD1044	в процессорах "System Q" и QnA не действует	Для выборочной трассировки	Шаг или время в момент выборочной трассировки	
D9049	SD1049		Рабочая область для программ на языке SFC	Адрес блока расширенных регистров файлов	
D9050	SD1050		Номер ошибки программы на языке SFC	Код ошибки, вызванной программой на языке SFC	
D9051	SD1051		Ошибка блока	Адрес блока, в котором возникла ошибка	
D9052	SD1052		Ошибка шага	Номер шага, в котором возникла ошибка	
D9053	SD1053		Ошибка перехода	Условия перехода, в которых возникла ошибка	
D9054	SD1054		Ошибка шага основной программы	Шаг основной программы, в котором возникла ошибка	
D9055	SD1055		SD812	Фиксация состояния	Номер шага фиксации состояния
D9060	SD1060	SD392	Версия программного обеспечения	Версия системного программного обеспечения	QnA
D9072	SD1072	в процессорах "System Q" и QnA не действует	Контроль коммуникации с компьютером	Контроль данных, обмениваемых через модуль Computer-Link	Q и QnA
D9081	SD1081	SD714	Количество свободных запросов коммуникации в области регистрации	Количество свободных запросов коммуникации в области регистрации	QnA

## Перечень специальных регистров и регистров диагностики

Специальный регистр процессора серии "А"	Конвертированный регистр диагностики	Эквивалентный регистр диагностики "System Q"/QnA	Название	Значение	Дейст. для:
D9085	SD1085	в процессорах "System Q" и QnA не действует	Регистр для установки времени интервала проверки	Предустановленное значение 10 с	Q и QnA
D9090	SD1090	в процессорах "System Q" и QnA не действует	AnN: диапазон адресов для подпрограммы программы микрокомпьютера AnA: не используется QnA: общее число специальных модулей	AnN: диапазон адресов для подпрограммы программы микрокомпьютера AnA: не используется QnA: общее число специальных модулей	Q и QnA
D9091	SD1091	в процессорах "System Q" и QnA не действует	Подробный код ошибки	Подробный код ошибки, выявленной функцией самодиагностики	
D9094	SD9094	SD251	Головной адрес заменяемого модуля ввода-вывода	Головной адрес заменяемого модуля ввода-вывода	
D9100	SD1100	—	Неисправен предохранитель модуля	Набор битов модуля, предохранитель которого неисправен.	
D9101	SD1101				
D9102	SD1102				
D9103	SD1103				
D9104	SD1104				
D9105	SD1105				
D9106	SD1106				
D9107	SD1107				
D9108	SD1108	—	Настройка контрольного времени для передачи шага	Настройка контрольного времени (от 1 до 255 с) в младшем байте и указание маркера ошибки в старшем байте.	
D9109	SD1109				
D9110	SD1110				
D9111	SD1111				
D9112	SD1112				
D9113	SD1113				
D9114	SD1114				
D9115	SD1115				
D9116	SD1116	—	Модуль ввода-вывода с ошибкой сверки	Набор битов модуля ввода-вывода с ошибкой сверки	
D9117	SD1117				
D9118	SD1118				
D9119	SD1119				
D9120	SD1120				
D9121	SD1121				
D9122	SD1122				
D9123	SD1123				
D9124	SD9124	SD63	Сумма маркеров ошибок	Сумма маркеров ошибок	



## Перечень специальных регистров и регистров диагностики

Специальный регистр процессора серии "А"	Конвертированный регистр диагностики	Эквивалентный регистр диагностики "System Q"/QnA	Название	Значение	Дейст. для:
D9125	SD9125	SD64	Номера маркеров ошибок	Номера маркеров ошибок	Q и QnA
D9126	SD9126	SD65			
D9127	SD9127	SD66			
D9128	SD9128	SD67			
D9129	SD9129	SD68			
D9130	SD9130	SD69			
D9131	SD9131	SD70			
D9132	SD9132	SD71			
D9200	SD1200	—	Результаты обработки LRDP	0 : нормальное завершение 2 : ошибка настройки команды LRDP 3 : сбой в станции, к которой осуществляется обращение 4 : выполнение команды LRDP в целевой станции не возможно	QnA
D9201	SD1201	—	Результаты обработки LWTP	0 : нормальное завершение 2 : ошибка настройки команды LWTP 3 : сбой в станции, к которой осуществляется обращение 4 : выполнение LWTP в целевой станции не возможно	
D9202	SD1202	—	Тип соединения локальной станции	Сохраняет условия для 16 адресов (1–16)	
D9203	SD1203	—		Сохраняет станции (17–32)	
D9241	SD1241	—		Сохраняет станции (33–48)	
D9242	SD1242	—		Сохраняет станции (49–64)	
D9204	SD1204	—	Состояние связи	0: передача данных по прямой петле 1: передача данных по обратной петле 2: передача данных в прямом и обратном направлении 3: петля только через прямую петлю 4: петля только через обратную петлю 5: передача данных не возможна	
D9205	SD1205	—	Станция, образующая петлю	Станция, через которую образуется прямая/обратная петля связи	
D9206	SD1206	—	Станция, образующая петлю	Станция, через которую образуется прямая/обратная петля связи	
D9207	SD1207	—	Время цикла в сети	Максимальное значение	
D9208	SD1208	—		Минимальное значение	
D9209	SD1209	—		Фактическое значение	
D9210	SD1210	—	Количество повторений	Сохраняет накопленное значение	
D9211	SD1211	—	Счетчик смены линии	Сохраняет накопленное значение	

## Перечень специальных регистров и регистров диагностики

Специальный регистр процессора серии "А"	Конвертированный регистр диагностики	Эквивалентный регистр диагностики "System Q"/QnA	Название	Значение	Дейст. для:
D9212	SD1212	—	Рабочее состояние локальной станции	Состояние станций 1–16	QnA
D9213	SD1213	—		Состояние станций 17–32	
D9214	SD1214	—		Состояние станций 33–48	
D9215	SD1215	—		Состояние станций 49–64	
D9216	SD1216	—	Распознавание ошибки локальной станции	Состояние станций 1–16	
D9217	SD1217	—		Состояние станций 17–32	
D9218	SD1218	—	Распознавание ошибки локальной станции	Состояние станций 33–48	
D9219	SD1219	—		Состояние станций 49–64	
D9220	SD1220	—	Несоответствие параметров локальной станции, ошибочное присвоение входов и выходов удаленной станции ввода-вывода	Состояние станций 1–16	
D9221	SD1221	—		Состояние станций 17–32	
D9222	SD1222	—		Состояние станций 33–48	
D9223	SD1223	—		Состояние станций 49–64	
D9224	SD1224	—	Инициализирующая передача в локальную станцию или удаленные станции ввода-вывода	Состояние станций 1–16	
D9225	SD1225	—		Состояние станций 17–32	
D9226	SD1226	—		Состояние станций 33–48	
D9227	SD1227	—		Состояние станций 49–64	
D9228	SD1228	—	Ошибка в удаленной станции ввода-вывода или локальной станции	Состояние станций 1–16	
D9229	SD1229	—		Состояние станций 17–32	
D9230	SD1230	—		Состояние станций 33–48	
D9231	SD1231	—		Состояние станций 49–64	
D9232	SD1232	—	Ошибка петли локальной или удаленной станции ввода-вывода	Состояние станций 1–8	
D9233	SD1233	—		Состояние станций 9–16	
D9234	SD1234	—		Состояние станций 17–24	
D9235	SD1235	—		Состояние станций 25–32	
D9236	SD1236	—		Состояние станций 33–40	
D9237	SD1237	—		Состояние станций 41–48	
D9238	SD1238	—		Состояние станций 49–56	
D9239	SD1239	—		Состояние станций 57–64	
D9240	SD1240	—		Счетчик ошибок передачи	Состояние станций 1–16
D9243	SD1243	—	Номер локальной станции или удаленной станции ввода-вывода	Состояние станций 17–32	
D9244	SD1244	—	Количество подключенных подчиненных станций в сети	Сохраняет сумму подчиненных станций	
D9245	SD1245	—	Счетчик ошибок передачи	Сохраняет сумму ошибок передачи	

## Перечень специальных регистров и регистров диагностики

Специальный регистр процессора серии "А"	Конвертированный регистр диагностики	Эквивалентный регистр диагностики "System Q"/QnA	Название	Значение	Дейст. для:
D9248	SD1248	—	Рабочее состояние локальных станций	Состояние станций 1–16	QnA
D9249	SD1249	—		Состояние станций 17–32	
D9250	SD1250	—		Состояние станций 33–48	
D9251	SD1251	—		Состояние станций 49–64	
D9252	SD1252	—	Ошибка в локальной станции	Состояние станций 1–16	
D9253	SD1253	—		Состояние станций 17–32	
D9254	SD1254	—		Состояние станций 33–48	
D9255	SD1255			Состояние станций 49–64	

## А.5.2 Специальные регистры (только MELSEC серии "А")

Специальные регистры – это регистры данных для специфических задач в центральном процессоре. За исключением регистров, обозначенных символом **❶**, в специальные регистры не могут записываться любые данные. Эти регистры можно подразделить на следующие виды:

- Специальные регистры, в которые данные автоматически записывает центральный процессор. Пользователь может только считывать (и сбрасывать) эти регистры.
- Специальные регистры, в которые данные могут быть записаны при определенных условиях.

Область применения специальных регистров в основной программе следует соответственно проверять.

В следующей таблице дан обзор специальных регистров и их назначения.

Перечень специальных регистров (MELSEC серии "А")

Адрес	Название	Значение	Описание	Центр. проц.
D9000	Неисправен предохранитель	Адрес модуля, в котором имеется неисправный предохранитель	Если в одном или нескольких модулях в сети обнаружена неисправность предохранителя, адрес первого распознанного модуля записывается в этот регистр. В случае нескольких модулей в регистр записывается самый низкий адрес (например, если неисправны предохранители в выходных модулях с диапазоном адресов от Y50 до 6F, в регистр D9000 записывается "50"). Эта проверка выполняется и для выходных модулей удаленных станций ввода-вывода.  Адрес модуля сохраняется в шестнадцатеричном виде. В таком же виде он отображается и в программаторе.  Содержимое регистра стирается, как только регистры D9100...D9107 устанавливаются на "0".	
D9002	Ошибка сверки модуля ввода-вывода	Адрес модуля, в отношении которого имеется ошибка сверки	Если после включения питания фактическое состояние модуля ввода-вывода отличается от заданного, то в регистр D9002 записывается адрес модуля, в отношении которого распознана ошибка сверки. Эта проверка выполняется и для выходных модулей удаленных станций ввода-вывода.  Адрес модуля сохраняется в шестнадцатеричном виде (формат сохранения адреса аналогичен регистру D9000).  Содержимое регистра стирается, как только регистры с D9116 по D9123 устанавливаются на "0".	
<b>❶</b> D9004	Ошибка в мастер-модуле MELSECNET/MINI-S3	Состояние ошибки	После возникновения ошибки в MELSECNET/MINI-S3 этот регистр сохраняет состояние определения ошибки в главном модуле сети MELSECNET/MINI-S3 (AJ71PT32(S3)).  Более подробная информация о значении набора битов и состоянии ошибки имеется в руководстве по сети MELSECNET/MINI-S3.	только для AnA, AnAS, AnU
<b>❶</b> D9005	Количество падений сетевого напряжения	Счетчик падений напряжения	При каждом падении напряжения более чем на 20 % от номинального напряжения во время работы значение этого регистра повышается на "1". Значение регистра имеет двоичный формат.	
<b>❶</b> D9008	Распознавание ошибки путем самодиагностики	Номер ошибки	Если центральный процессор распознал ошибку с помощью функции самодиагностики, в этот регистр записывается соответствующий код ошибки (в двоичном виде).	только для AnS, A2C

Перечень специальных регистров (MELSEC серии "А")

Адрес	Название	Значение	Описание	Центр. проц.
D9009	Распознавание маркера ошибки	Маркер ошибки F для обозначения внешней ошибки	Сразу после установки маркера ошибки (от F0 до F255) командой OUT F или SET F номер первой распознанной ошибки записывается в двоичном виде в регистр D9009.  Содержимое регистра можно стереть с помощью команды RST F или LEDR. Если имеются другие номера F маркеров ошибок, команда стирания вызывает запись очередного номера ошибки в регистр D9009.	(кроме АЗН, АЗМ, АЗА, АЗН)
			Сразу после установки маркера ошибки (от F0 до F255) командой OUT F или SET F номер первой распознанной ошибки записывается в двоичном виде в регистр D9009.  Содержимое регистра можно стереть с помощью команды RST F или LEDR, либо включив выключатель „INDICATOR RESET“. Если имеются и другие номера маркеров ошибки, то в результате команды стирания в регистр D9009 записывается очередной номер.	только для АЗН, АЗМ, АЗА, АЗН
D9010	Номер ошибочного шага программы	Номер шага программы, в котором возникла ошибка	При возникновении ошибки во время выполнения прикладной команды этот регистр сохраняет в виде двоичного значения номер шага программы, в котором возникла ошибка. После каждого повторного возникновения ошибки содержимое D9010 обновляется.	(кроме АЗН, АЗМ)
D9011	Номер ошибочного шага программы	Номер шага программы, в котором возникла ошибка	При возникновении ошибки во время выполнения прикладной команды этот регистр сохраняет в виде двоичного значения номер шага программы, в котором возникла ошибка. Так как сохранение в D9011 происходит в тот же момент, в который устанавливается M9011, сброс содержимого регистра возможен лишь после сброса маркера M9011.	
D9014	Обозначение обработки входов и выходов	Вид обработки	Обработка входов и выходов выполняется в соответствии с сохраненным значением регистра. 0: Непосредственная обработка входных/выходных сигналов 1: Входы обрабатываются после внутреннего отображения; выходы обрабатываются непосредственно 3: Входы и выходы обрабатываются после внутреннего отображения	(кроме АЗН, АЗМ, АН)
D9015	Рабочее состояние центрального процессора	Режим	<p>Режим, в котором находится центральный процессор, сохраняется в D9015 следующим образом:</p> <p>*Режим PAUSE должен быть деблокирован с помощью маркера M9040. Если M9040 не установлен, переключение центрального процессора из режима RUN в режим PAUSE не возможно.</p>	


Перечень специальных регистров (MELSEC серии "А")

Адрес	Название	Значение	Описание	Центр. проц.
D9016	Настройка памяти	0: ROM 1: RAM 2: EEPROM	В D9016 сохраняется информация о выбранном запоминающем устройстве. Значение 0, 1 или 2 записывается в двоичном виде.	только для A1, A1N
	Обозначение программы	0: Основная программа (ROM) 1: Основная программа (RAM) 2: Подпрограмма (RAM)	D9016 служит в качестве обозначения памяти, в которой хранится обрабатываемая в данный момент основная программа. Значение 0, 1 или 2 записывается в двоичном виде.  Значение 2 возможно только в случае процессоров A3N, A3M, A3A и A3H.	(кроме A1, A1N)
D9017	Время цикла	Минимальное время цикла (единица: 10 мс)	В D9017 в двоичном виде сохраняется минимальное время цикла программы. Если фактическое время цикла меньше значения, сохраненного в D9017, то после выполнения команды END прежнее значение перезаписывается новым.	
D9018	Время цикла	Время цикла (единица: 10 мс)	В D9018 в двоичном виде сохраняется время цикла программы. При этом прежнее значение заменяется новым после каждой обработки команды END.	
D9019	Время цикла	Максимальное время цикла (единица: 10 мс)	В D9019 в двоичном виде сохраняется максимальное время цикла программы. Если фактическое время цикла больше значения, сохраненного в D9017, то после выполнения команды END прежнее значение перезаписывается новым.	
② D9020	Постоянное время цикла программы	Заданное значение постоянного времени цикла	В D9020 записывается заданное значение постоянного времени цикла основной программы. Единица значения регистра равна 10 мс.  0:            регистр не установлен от 1 до 200: программа выполняется с постоянными интервалами длительностью (значение регистра) x 10 мс.	(кроме серии "А")
D9021	Время цикла	Время цикла (единица: 1 мс)	Время цикла программы сохраняется в двоичном виде. При этом прежнее значение заменяется новым после каждой обработки команды END.	только для AnA, AnAS, AnU
D9022	1-секундный счетчик	Возрастающий и убывающий счет	При начале режима RUN центрального процессора счетчик начинает счет с секундным тактом. Счетчик считает от 0 до 32767 вверх, а затем перескакивает на -32768 и считает до 0.	только для AnA, AnAS, AnU
② D9025	Данные часов	Год и месяц	Этот регистр сохраняет год и месяц в двоично-десятичном виде следующим образом: Пример: 1992, июль = <b>9207</b>	только для AnA, AnAS, AnU, AnN, A1S
② D9026		День и час	Этот регистр сохраняет день и час в двоично-десятичном виде следующим образом: Пример: 31-е число, 10 часов = <b>3110</b>	только для AnA, AnAS, AnU, AnN, AnS
② D9027		Минута и секунда	Этот регистр сохраняет минуту и секунду в двоично-десятичном виде следующим образом:  Пример: 35 минут, 48 с = <b>3548</b>	только для AnA, AnAS, AnU, AnN, AnS
② D9028		День недели	Этот регистр сохраняет день недели в двоично-десятичном виде следующим образом (0 = воскресенье, 1 = понедельник, 2 = вторник и т. п.):	только для AnA, AnAS, AnU, AnN, AnS

Перечень специальных регистров (MELSEC серии "А")

Адрес	Название	Значение	Описание	Центр. проц.																																
D9021	Номер станции удаленных специальных модулей	от 1 до 61	<p>В эти регистры записываются номера станций удаленных специальных модулей, соединенных с процессором А2С. Последовательность записей не обязательно совпадает с действительной последовательностью модулей в сети.</p> <p>Структура данных:</p> <table border="1"> <tr> <td>D9021</td> <td>область для удал. спец. модуля №1</td> </tr> <tr> <td>D9022</td> <td>область для удал. спец. модуля №2</td> </tr> <tr> <td></td> <td>:</td> </tr> <tr> <td></td> <td>:</td> </tr> <tr> <td></td> <td>:</td> </tr> <tr> <td></td> <td>:</td> </tr> <tr> <td>D9032</td> <td>область для удал. спец. модуля №12</td> </tr> <tr> <td>D9033</td> <td>область для удал. спец. модуля №13</td> </tr> <tr> <td>D9034</td> <td>область для удал. спец. модуля №14</td> </tr> </table>	D9021	область для удал. спец. модуля №1	D9022	область для удал. спец. модуля №2		:		:		:		:	D9032	область для удал. спец. модуля №12	D9033	область для удал. спец. модуля №13	D9034	область для удал. спец. модуля №14	только для А2С														
D9021				область для удал. спец. модуля №1																																
D9022				область для удал. спец. модуля №2																																
				:																																
				:																																
				:																																
				:																																
D9032				область для удал. спец. модуля №12																																
D9033				область для удал. спец. модуля №13																																
D9034				область для удал. спец. модуля №14																																
D9022																																				
D9023																																				
D9024																																				
D9025																																				
D9026																																				
D9027																																				
D9028																																				
D9029																																				
D9030																																				
D9031																																				
D9032																																				
D9033																																				
D9034																																				
D9035	Атрибуты удаленных специальных модулей	0: стандартный протокол 1: без протокола	<p>Регистр содержит атрибуты каждого специального модуля, имеющегося в сети А2С, в виде значений „0” и „1” (более подробно разъяснение имеется в описании аппаратуры А2С).</p> <p>Структура данных:</p> <table border="1"> <tr> <td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td> </tr> </table> <p>↑ индикация модуля с неисправным предохранителем</p> <p>Бит b0 указывает атрибут для специального модуля 1, бит b1 – для модуля 2, бит b3 – для модуля 3 и т. д.</p>	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	только для А2С
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																					
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0																					
D9036	Суммарное количество станций	от 1 до 64	В этот регистр записывается общее количество (от 1 до 64) станций удаленных модулей ввода-вывода и специальных модулей, и находящихся в сети А2С.	только для А2С																																
D9036	Адреса операндов расширенных регистров файлов	Адреса операндов	Адреса операндов расширенных регистров файлов для непосредственного считывания и записи указываются в двоичном виде в двойном слове D9036 и D9037. Адреса должны быть указаны непрерывно, начиная с R0 блока № 0. Более подробная информация имеется в описании аппаратуры MELSEC серии "А".	только для AnA, AnAS, AnU																																
D9037																																				
D9038	Приоритет светодиодной индикации ошибок	Уровень приоритета от 1 до 4	<p>В эти регистры записывается приоритет светодиодной индикации в зависимости от номера ошибки. Уровень приоритета можно задать, изменив последовательность битов в регистре.</p> <p>Структура данных:</p> <table border="1"> <tr> <td>приоритет 4</td> <td>приоритет 3</td> <td>приоритет 2</td> <td>приоритет 1</td> </tr> <tr> <td></td> <td></td> <td>приоритет 0</td> <td></td> </tr> </table>	приоритет 4	приоритет 3	приоритет 2	приоритет 1			приоритет 0		только для AnA, AnAS, AnU, А2С, AnS																								
приоритет 4		приоритет 3		приоритет 2	приоритет 1																															
		приоритет 0																																		
D9039	Уровень приоритета от 5 до 7	Более подробная информация имеется в описании аппаратуры серии А2С, AnA или AnU.																																		
D9055	Номер шага фиксации состояния	Номер шага	Сохраняет номер шага после выполнения фиксации состояния.	только для AnA, AnAS, AnU																																

## Перечень специальных регистров (MELSEC серии "А")

Адрес	Название	Значение	Описание	Центр. проц.																																																																																					
D9056 до D9059	Обозначение ошибочной станции	0: нормальное состояние 1: ошибка	Если возник сбой коммуникации с удаленным модулем, то в этом регистре бит соответствующей станции устанавливается на „1“. Бит устанавливается на „1“ лишь в том случае, если связь не удалось установить и после количества попыток повторения, заданного в D9174. Бит остается установленным и в том случае, если неполадка устранена и связь со станцией налажена.  Структура данных: <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td></td> <td>b15</td> <td>b14</td> <td>b13</td> <td>b12</td> <td>b11</td> <td>b10</td> <td>b9</td> <td>b8</td> <td>b7</td> <td>b6</td> <td>b5</td> <td>b4</td> <td>b3</td> <td>b2</td> <td>b1</td> <td>b0</td> </tr> <tr> <td>D9056</td> <td>16</td> <td>15</td> <td>14</td> <td>13</td> <td>12</td> <td>11</td> <td>10</td> <td>9</td> <td>8</td> <td>7</td> <td>6</td> <td>5</td> <td>4</td> <td>3</td> <td>2</td> <td>1</td> </tr> <tr> <td>D9057</td> <td>32</td> <td>31</td> <td>30</td> <td>29</td> <td>28</td> <td>27</td> <td>26</td> <td>25</td> <td>24</td> <td>23</td> <td>22</td> <td>21</td> <td>20</td> <td>19</td> <td>18</td> <td>17</td> </tr> <tr> <td>D9058</td> <td>48</td> <td>47</td> <td>46</td> <td>45</td> <td>44</td> <td>43</td> <td>42</td> <td>41</td> <td>40</td> <td>39</td> <td>38</td> <td>37</td> <td>36</td> <td>35</td> <td>34</td> <td>33</td> </tr> <tr> <td>D9059</td> <td>64</td> <td>63</td> <td>62</td> <td>61</td> <td>60</td> <td>59</td> <td>58</td> <td>57</td> <td>56</td> <td>55</td> <td>54</td> <td>53</td> <td>52</td> <td>51</td> <td>50</td> <td>49</td> </tr> </table> Номера станций  Соответствующий бит устанавливается на „1“ даже в том случае, если неисправен предохранитель удаленного модуля ввода-вывода.		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9056	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	D9057	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	D9058	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	D9059	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	только для A2C
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																									
D9056	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1																																																																									
D9057	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17																																																																									
D9058	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33																																																																									
D9059	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49																																																																									
D9061	Ошибка коммуникации	0: нормальное состояние 1: ошибка при инициализации 2: неисправность проводки	После установки маркера M9061 (ошибка коммуникации с удаленным модулем) в D9061 записывается номер кода ошибки.  Значение „1“ означает, что в данных инициализации имеется ошибка (неправильная сумма станций или неправильное количество попыток повторения).  В случае „2“ имеется дефект проводки либо не включен один из удаленных модулей.	только для A2C																																																																																					
D9072	Контроль коммуникации с компьютером	Контроль обмена данными через модуль Computer-Link	Контрольный регистр самотестирования модуля Computer-Link для записи, считывания и передачи данных.	только для AnA, AnAS, AnU																																																																																					
D9073	Данные часов	Год и месяц	Функция аналогична специальному регистру D9035.	только для A2C C24(-PRF)																																																																																					
D9074	Данные часов	День и час	Функция аналогична специальному регистру D9036.																																																																																						
D9075	Данные часов	Минута и секунда	Функция аналогична специальному регистру D9037.																																																																																						
D9076	Данные часов	День недели	Функция аналогична специальному регистру D9038.																																																																																						
D9081	Количество запросов на коммуникацию с удаленным специальным модулем	от 0 до 32	В D9081 сохраняется количество запросов на обмен данными с удаленным специальным модулем с помощью команд FROM и TO.  Каждый раз по окончании коммуникации со специальным модулем количество запросов уменьшается на 1.	только для AnA, AnAS, AnU, A2C																																																																																					
D9082	Номер последней станции в сети	Номер последней станции	В регистр D9082 записывается номер станции последнего удаленного модуля, подключенного к сети A2C.	только для A2C																																																																																					
D9090	Диапазон адресов для подпрограммы программы микрокомпьютера	В зависимости от применяемой программы микрокомпьютера	Более подробные указания имеются в программном обеспечении соответствующего микрокомпьютера.	(кроме AnA, AnAS, AnU)																																																																																					
D9091	Ошибка команды	Код ошибки команды	В этом регистре сохраняется код, указывающий причину ошибки команды.	только для AnA, AnAS, AnU																																																																																					
D9091	Код ошибки для программы микрокомпьютера	В зависимости от используемой программы микрокомпьютера	Более подробные указания имеются в программном обеспечении соответствующего микрокомпьютера.	(кроме AnA, AnAS, AnU)																																																																																					
 D9094	Адрес заменяемого модуля ввода-вывода	Начальный адрес модуля ввода-вывода	Регистр D9094 сохраняет в двоичном виде первые два разряда начального адреса (например, входной модуль X2FO = H2F) модуля ввода-вывода, который был вынут из монтажной шины или вставлен в нее во время режима онлайн.	только для AnN, AnA, AnAS, AnU																																																																																					



Перечень специальных регистров (MELSEC серии "А")

Адрес	Название	Значение	Описание	Центр. проц.				
<p>①</p> <p>D9100</p>	Неисправен предохранитель модуля	Набор битов, отображающий адреса модулей	<p>Если в модуле ввода-вывода обнаружен неисправный предохранитель, то в наборе битов регистра D9100 соответствующий этому модулю бит устанавливается на „1“. Присвоение битов осуществляется с помощью параметров.</p> <p style="text-align: center;">                     b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0                      D9100 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0                 </p> <p style="text-align: center;">↑ индикация модуля с неисправным предохранителем</p> <p>После замены предохранителя и возобновления работы модуля установленный бит следует сбрасывать с помощью основной программы.</p>	только для A1S				
<p>①</p> <p>D9100</p>	Неисправен предохранитель модуля	Набор битов, отображающий адреса модулей	<p>Если в модуле обнаружен неисправный предохранитель, адрес этого модуля записывается в набор битов D9100...D9107 (в единицах по 16). Соответствие между адресами ввода-вывода и набором битов предусмотрено в параметрах.</p> <p style="text-align: center;">                     b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0                      D9100 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0                      D9101 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0                      : : : : : : : : : : : : : : : : : :                      D9107 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0                 </p> <p style="text-align: center;">↑ индикация модуля с неисправным предохранителем</p> <p>После восстановления нормального состояния соответствующие биты не сбрасываются на „0“ автоматически. Поэтому сброс должен осуществляться с помощью основной программы.</p>	<p>(кроме A1S, в процессорах AnS только от D9100 до D9103)</p>				
<p>①</p> <p>D9101</p>								
<p>①</p> <p>D9102</p>								
<p>①</p> <p>D9103</p>								
<p>①</p> <p>D9104</p>								
<p>①</p> <p>D9105</p>								
<p>①</p> <p>D9106</p>								
<p>①</p> <p>D9107</p>								
<p>①</p> <p>D9116</p>	Модули ввода-вывода с ошибкой сверки	Набор битов, отображающий адреса модулей	<p>Если текущее состояние модуля ввода-вывода отличается от заданного состояния после включения напряжения питания, то в наборе битов D9116 присвоенный этому модулю бит устанавливается на 1. Присвоение битов осуществляется с помощью параметров.</p> <p>После восстановления нормального состояния бит следует сбрасывать на 0 с помощью основной программы.</p>	только для A1S				
<p>D9116</p> <p>D9117</p> <p>D9118</p> <p>D9119</p> <p>D9120</p> <p>D9121</p> <p>D9122</p>	Модули ввода-вывода с ошибкой сверки	Набор битов, отображающий адреса модулей	<p>Если текущее состояние модуля ввода-вывода отличается от заданного состояния после включения напряжения питания, то в наборе битов D9116...D123 соответствующий этому модулю бит устанавливается на „1“. Набор битов, составленный в единицах по 16 битов, соответствует адресам ввода-вывода, предварительно установленным в параметрах.</p> <p style="text-align: center;">                     b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0                      D9116 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0                      D9117 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0                      : : : : : : : : : : : : : : : : : :                      D9123 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0                 </p> <p style="text-align: center;">↑ индикация модуля с неисправным предохранителем</p> <p>После восстановления нормального состояния соответствующие биты не сбрасываются на „0“ автоматически. Поэтому сброс должен осуществляться с помощью основной программы.</p>	<p>(кроме A1S, в процессорах AnS только от D9116 до D9119)</p>				
<p>D9123</p>								
<p>D9124</p>					Общее количество маркеров ошибки	Общее количество маркеров ошибки	<p>Если команда OUT F или SET F устанавливает маркер ошибки (F0...F255), то значение в D9124 повышается на „1“. При выполнении команды RST F или SET F значение этого регистра снова уменьшается на „1“.</p> <p>Суммарное количество маркеров ошибки сохраняется в D9124 в двоичном виде. Поэтому максимальное значение регистра D9124 равно 8.</p>	

Перечень специальных регистров (MELSEC серии "А")

Адрес	Название	Значение	Описание	Центр. проц.																																																																																																																																		
D9125 — D9132	Номер маркера ошибки		<p>Если команда OUT F или SET F установила один из маркеров ошибки F0...F255, адрес установленного маркера ошибки записывается в двоичном виде в регистры с D9125 по D9132.</p> <p>Адрес маркера ошибки, сбрасываемый командой RST F, стирается из области регистра. Затем в этот регистр передается содержимое следующих регистров данных.</p> <p>При выполнении команды LEDR содержимое регистров с D9125 по D9132 смещается на один бит вверх.</p> <p>Если имеется более 8 сообщений об ошибках, то 9-й маркер ошибки в регистрах D9125...D9132 не сохраняется.</p> <div style="text-align: center;"> <p>SET SET SET SET SET SET SET SET SET SET F50 F25 F19 F25 F15 F70 F65 F38 F110 F151 F210 LEDR</p> <p>→ → → → → → → → → → → → → →</p> <table border="1"> <tr><td>D9009</td><td>0</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>99</td></tr> <tr><td>D9124</td><td>0</td><td>1</td><td>2</td><td>3</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>8</td></tr> <tr><td>D9125</td><td>0</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>50</td><td>99</td></tr> <tr><td>D9126</td><td>0</td><td>0</td><td>25</td><td>25</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>99</td><td>15</td></tr> <tr><td>D9127</td><td>0</td><td>0</td><td>0</td><td>99</td><td>0</td><td>15</td><td>15</td><td>15</td><td>15</td><td>15</td><td>15</td><td>70</td></tr> <tr><td>D9128</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>70</td><td>70</td><td>70</td><td>70</td><td>70</td><td>70</td><td>65</td></tr> <tr><td>D9129</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>65</td><td>65</td><td>65</td><td>65</td><td>65</td><td>38</td></tr> <tr><td>D9130</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>38</td><td>38</td><td>38</td><td>38</td><td>110</td></tr> <tr><td>D9131</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>110</td><td>110</td><td>110</td><td>151</td></tr> <tr><td>D9132</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>151</td><td>151</td><td>210</td></tr> </table> </div>	D9009	0	50	50	50	50	50	50	50	50	50	50	99	D9124	0	1	2	3	2	3	4	5	6	7	8	8	D9125	0	50	50	50	50	50	50	50	50	50	50	99	D9126	0	0	25	25	99	99	99	99	99	99	99	15	D9127	0	0	0	99	0	15	15	15	15	15	15	70	D9128	0	0	0	0	0	70	70	70	70	70	70	65	D9129	0	0	0	0	0	0	65	65	65	65	65	38	D9130	0	0	0	0	0	0	0	38	38	38	38	110	D9131	0	0	0	0	0	0	0	0	110	110	110	151	D9132	0	0	0	0	0	0	0	0	0	151	151	210	
D9009	0	50	50	50	50	50	50	50	50	50	50	99																																																																																																																										
D9124	0	1	2	3	2	3	4	5	6	7	8	8																																																																																																																										
D9125	0	50	50	50	50	50	50	50	50	50	50	99																																																																																																																										
D9126	0	0	25	25	99	99	99	99	99	99	99	15																																																																																																																										
D9127	0	0	0	99	0	15	15	15	15	15	15	70																																																																																																																										
D9128	0	0	0	0	0	70	70	70	70	70	70	65																																																																																																																										
D9129	0	0	0	0	0	0	65	65	65	65	65	38																																																																																																																										
D9130	0	0	0	0	0	0	0	38	38	38	38	110																																																																																																																										
D9131	0	0	0	0	0	0	0	0	110	110	110	151																																																																																																																										
D9132	0	0	0	0	0	0	0	0	0	151	151	210																																																																																																																										
D9133 — D9140	Информация об удаленных модулях, применяемых в системе A2C	<p>00: под этим номером станции не находится модуль ввода-вывода или специальный модуль, либо не выполнена инициализация</p> <p>01: входной модуль A2C или специальный модуль</p> <p>10: выходной модуль A2C</p> <p>11: удаленный специальный модуль</p>	<p>Регистры с D9133 по D9140 содержат информацию об удаленных модулях, имеющихся в системе A2C. Соответствующая информация сопоставлена номерам станций модулей и отображена в виде 2-битных данных.</p> <p>Структура данных:</p> <table border="1"> <tr> <td>D9133</td> <td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>D9133</td> <td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>D9134</td> <td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>D9135</td> <td>16</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td></td> <td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td> </tr> <tr> <td>D9139</td> <td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>D9140</td> <td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table> <p style="text-align: center;">— Номера станций —</p>	D9133	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9133	8	7	6	5	4	3	2	1									D9134	16	15	14	13	12	11	10	9									D9135	16	23	22	21	20	19	18	17										:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	D9139	56	55	54	53	52	51	50	49									D9140	64	63	62	61	60	59	58	57									только для A2C											
D9133	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																																																																						
D9133	8	7	6	5	4	3	2	1																																																																																																																														
D9134	16	15	14	13	12	11	10	9																																																																																																																														
D9135	16	23	22	21	20	19	18	17																																																																																																																														
	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:	:																																																																																																																						
D9139	56	55	54	53	52	51	50	49																																																																																																																														
D9140	64	63	62	61	60	59	58	57																																																																																																																														

Перечень специальных регистров (MELSEC серии "А")

Адрес	Название	Значение	Описание	Центр. проц.																																																																	
D9141 — D9172	Счетчик повторений	Количество повторений	<p>В регистры с D9141 по D9172 для каждой станции записывается количество повторных обращений из центрального процессора, сделанных при сбое коммуникации с удаленной станцией (количество попыток повторения устанавливается в D9174).</p> <p>Если безошибочную коммуникацию снова удалось восстановить, соответствующий бит устанавливается на „0“.</p> <p>Номера станций имеют следующую структуру:</p> <table border="1" style="margin-left: 40px;"> <tr> <td></td> <td style="text-align: center;">b15</td> <td style="text-align: center;">-</td> <td style="text-align: center;">b8</td> <td style="text-align: center;">b7</td> <td style="text-align: center;">-</td> <td style="text-align: center;">b0</td> </tr> <tr> <td>D9141</td> <td colspan="2" style="text-align: center;">станция 2</td> <td colspan="3"></td> <td style="text-align: center;">станция 1</td> </tr> <tr> <td>D9142</td> <td colspan="2" style="text-align: center;">станция 4</td> <td colspan="3"></td> <td style="text-align: center;">станция 3</td> </tr> <tr> <td>D9143</td> <td colspan="2" style="text-align: center;">станция 6</td> <td colspan="3"></td> <td style="text-align: center;">станция 5</td> </tr> <tr> <td style="text-align: center;">↓</td> <td style="text-align: center;">⋮</td> <td></td> <td style="text-align: center;">⋮</td> <td></td> <td style="text-align: center;">⋮</td> <td></td> </tr> <tr> <td>D9171</td> <td colspan="2" style="text-align: center;">станция 62</td> <td colspan="3"></td> <td style="text-align: center;">станция 61</td> </tr> <tr> <td>D9172</td> <td colspan="2" style="text-align: center;">станция 64</td> <td colspan="3"></td> <td style="text-align: center;">станция 63</td> </tr> </table> <p>Счетчик повторений станции состоит из 8 бит. Первый бит показывает состояние (0 = нормальное, 1 = ошибка), а остальные 7 битов указывают количество попыток доступа.</p> <table border="1" style="margin-left: 40px;"> <tr> <td style="text-align: center;">b(n+7)</td> <td style="text-align: center;">b(n+6)</td> <td style="text-align: center;">b(n+5)</td> <td style="text-align: center;">b(n+4)</td> <td style="text-align: center;">b(n+3)</td> <td style="text-align: center;">b(n+2)</td> <td style="text-align: center;">b(n+1)</td> <td style="text-align: center;">b(n+0)</td> </tr> <tr> <td style="text-align: center;">0/1</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </table> <p style="text-align: center; margin-left: 40px;">Количество повторных обращений</p> <p>Значение n определяется номером станции модуля.                      1, 3, 5, ..., 61, 63 нечетная станция: от b0 до b7 (n = 0)                      2, 4, 6, ..., 62, 64 четная станция: от b8 до b15 (n = 8)</p>		b15	-	b8	b7	-	b0	D9141	станция 2					станция 1	D9142	станция 4					станция 3	D9143	станция 6					станция 5	↓	⋮		⋮		⋮		D9171	станция 62					станция 61	D9172	станция 64					станция 63	b(n+7)	b(n+6)	b(n+5)	b(n+4)	b(n+3)	b(n+2)	b(n+1)	b(n+0)	0/1								только для А2С
	b15	-	b8	b7	-	b0																																																															
D9141	станция 2					станция 1																																																															
D9142	станция 4					станция 3																																																															
D9143	станция 6					станция 5																																																															
↓	⋮		⋮		⋮																																																																
D9171	станция 62					станция 61																																																															
D9172	станция 64					станция 63																																																															
b(n+7)	b(n+6)	b(n+5)	b(n+4)	b(n+3)	b(n+2)	b(n+1)	b(n+0)																																																														
0/1																																																																					
D9173	Режим	Числовой код режима: 0: автоматическое возобновление 1: без автоматического возобновления 2: прерывание коммуникации при ошибке в режиме онлайн 3: аппаратный тест	Настроенный режим Режим 0 (автоматическая возобновление): Станции, имеющие ошибки, исключаются из сети. Коммуникация с прочими станциями продолжается. После устранения неисправности работа с соответствующей станцией автоматически возобновляется. Режим 1 (без автоматического возобновления): Станции, имеющие ошибки, исключаются из сети. Коммуникация с прочими станциями продолжается. После устранения ошибки работа с соответствующей станцией может быть возобновлена лишь после перезапуска модуля станции. Режим 2 (прерывание коммуникации при возникновении ошибки онлайн): Если в каком-либо модуле возникла ошибка, прерывается коммуникация со всеми станциями. После устранения ошибки работа с соответствующей станцией может быть возобновлена лишь после перезапуска модуля станции. Режим 3 (аппаратный тест): Проверка аппаратуры и соединительных кабелей																																																																		
D9174	Количество повторных обращений	Количество повторений (от 0 до 32)	<p>В регистр D9174 записывается количество повторных обращений, которое должно выполняться при сбое связи с удаленным модулем (стандартное значение: 5).</p> <p>Если коммуникация с удаленным модулем не возможна и после указанного здесь количества повторных попыток, выводится сообщение об ошибке.</p>																																																																		
D9175	Счетчик повторений при наличии ошибки в сети	Количество повторений	<p>При сбое связи с удаленным модулем (превышение времени), количество попыток повторения сохраняется в регистре D9175. После успешного возобновления связи с удаленными модулями значение регистра снова устанавливается на „0“.</p>																																																																		

Перечень специальных регистров (MELSEC серии "А")

Адрес	Название	Значение	Описание	Центр. проц.																																																																																				
D9180 — D9193	Код ошибки удаленных специальных модулей	Код ошибки (0: нормальное состояние)	<p>После установки маркера M9060 (распознавание ошибки) код ошибки удаленного специального модуля записывается в регистры D9180...D9193.</p> <p>Номера станций удаленных специальных модулей записываются в регистры с D9021 по D9034. Области памяти для кодов ошибок подразделены следующим образом.</p> <table border="1" style="margin-left: 40px;"> <tr><td>D9180</td><td>удаленный специальный модуль n° 1</td></tr> <tr><td>D9181</td><td>удаленный специальный модуль n° 2</td></tr> <tr><td>D9182</td><td>удаленный специальный модуль n° 3</td></tr> <tr><td>↓</td><td>:</td></tr> <tr><td>D9192</td><td>удаленный специальный модуль n° 13</td></tr> <tr><td>D9193</td><td>удаленный специальный модуль n° 14</td></tr> </table> <p>Код ошибки стирается при переключении выключателя с ключом из положения "STOP" в положение "RUN" (стираются все регистры) или при включении сигнала Yn4 удаленных специальных модулей.</p>	D9180	удаленный специальный модуль n° 1	D9181	удаленный специальный модуль n° 2	D9182	удаленный специальный модуль n° 3	↓	:	D9192	удаленный специальный модуль n° 13	D9193	удаленный специальный модуль n° 14	только для A2C																																																																								
D9180	удаленный специальный модуль n° 1																																																																																							
D9181	удаленный специальный модуль n° 2																																																																																							
D9182	удаленный специальный модуль n° 3																																																																																							
↓	:																																																																																							
D9192	удаленный специальный модуль n° 13																																																																																							
D9193	удаленный специальный модуль n° 14																																																																																							
D9196 — D9199	Обозначение неисправной станции	0: нормальное состояние 1: ошибка	<p>Если доступ к станции не возможен и после указанного в D9174 количества повторных попыток, сопоставленный этой станции бит в регистрах D9196...D9199 устанавливается на „1“. Этот бит снова сбрасывается на „0“, если неполадка устранена и было выбрано автоматическое возобновление (D9173).</p> <p>Структура данных:</p> <table border="1" style="margin-left: 40px;"> <tr> <td></td> <td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>D9196</td> <td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td> </tr> <tr> <td>D9197</td> <td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td> </tr> <tr> <td>D9198</td> <td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td> </tr> <tr> <td>D9199</td> <td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td> </tr> </table> <p>Соответствующий бит устанавливается на „1“ также в том случае, если неисправен предохранитель удаленного модуля ввода-вывода.</p>		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9196	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	D9197	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	D9198	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	D9199	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																								
D9196	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1																																																																								
D9197	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17																																																																								
D9198	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33																																																																								
D9199	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49																																																																								

**ПРИМЕЧАНИЯ** Сброс специальных регистров происходит после отключения питания контроллера или переключения выключателя с ключом в положение "LATCH CLEAR" или "RESET".

Переключение выключателя с ключом в положение "STOP" не вызывает сброса регистров. Фактические состояния сохраняются.

Содержимое специальных регистров, обозначенных сноской ❶, сохраняется и после восстановления нормального состояния. Сбросить эти специальные регистры можно следующим образом:

- вставив в основную программу строки, сбрасывающие специальный регистр с помощью команды RST при определенном входном условии,
- с помощью программатора.
- сбросив центральный процессор путем переключения выключателя с ключом на модуле центрального процессора в положение "RESET".

Специальные регистры, обозначенные сноской ❷, устанавливаются и сбрасываются только с помощью основной программы.

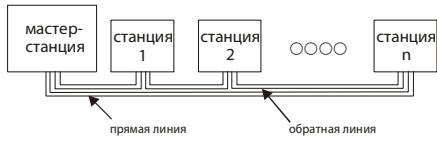
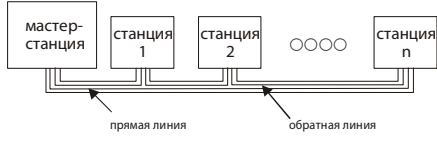
Специальные регистры, обозначенные сноской ❸, устанавливаются и сбрасываются в тестовом режиме с помощью программатора.

### А.5.3 Обзор специальных регистров связи (только MELSEC серии "А")

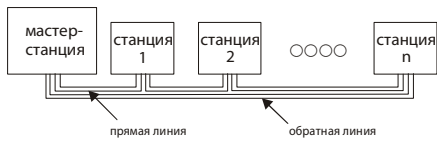
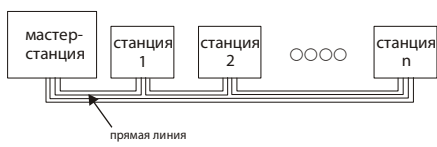
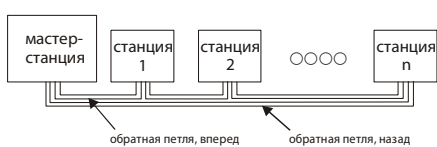
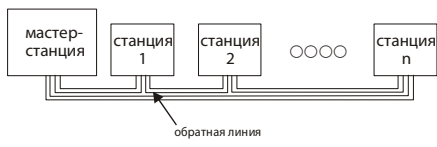
Эти специальные регистры устанавливаются и сбрасываются в зависимости от различных факторов во время обмена данными по коммуникационной сети. В этих регистрах сохраняется информация об определенных состояниях коммуникации или сбоях в сети. Ошибки в сети или неисправных станциях можно определить путем считывания регистров связи.

Обработка специальных регистров связи зависит от того, в какой станции находится центральный процессор – мастер-станции или локальной станции.

Специальные регистры связи в мастер-станции

Адрес	Название	Значение	Описание
D9200	Результат обработки команды LRDP	0: в порядке (норм. состояние) 2: ошибка в команде 3: ошибка в целевой станции 4: выполнение не возможно	Этот регистр сохраняет результат выполнения команды LRDP (считывание словного операнда).  0: Команда выполнена без ошибок. 2: Неправильно объявлена константа или исходные/целевые данные. 3: Станция, к которой обращается команда, является удаленной станцией.
D9201	Результат обработки команды LWTP	0: в порядке (норм. состояние) 2: ошибка в команде 3: ошибка в целевой станции 4: выполнение не возможно	Этот регистр сохраняет результат выполнения команды LWTP (запись в словный операнд).  0: Команда выполнена без ошибок. 2: Неправильно объявлена константа или исходные/целевые данные. 3: Станция, к которой обращается команда, является удаленной станцией.
D9202	Тип связи локальной станции (см. также D9241, D9242)	Состояние станций 1...16	
D9203		Состояние станций 17...32	
D9204 (см. также следующую страницу)	Состояние связи	0: Передача данных через прямую петлю 1: Передача данных через обратную петлю 2: Передача данных в прямом и обратном направлении 3: Петля только через прямую петлю 4: Петля только через обратную петлю 5: Передача данных не возможна  Примечание: Для MELSECNET/B возможны только варианты 0 и 5	В этом регистре сохраняются текущие данные состояния сетевой коммуникации.  – передача данных по прямой петле   – передача данных по обратной петле 

Специальные регистры связи в мастер-станции

Адрес	Название	Значение	Описание
D9204	Состояние связи		<p>– передача данных в прямом и обратном направлении</p> 
			<p>– передача данных только по прямой линии</p> 
			<p>– передача данных только по обратной линии</p> 
D9205*	Станция, выполняющая петлевой тест	Станция, через которую проходит прямая линия	Эти регистры данных сохраняют адрес локальной станции или удаленной станции ввода-вывода, через которую проходит прямая линия.
D9206*	Станция, выполняющая петлевой тест	Станция, через которую проходит обратная линия	 <p>В вышеприведенном примере в регистре D9205 сохраняется „1“, а в D9206 – „3“. Эти значения сохраняются даже в том случае, если было снова восстановлено нормальное состояние коммуникации. Сбрасывать значения регистров следует с помощью основной программы или путем сброса центрального процессора.</p>
D9207*	Время цикла в сети	Максимальное время цикла	<p>В этих регистрах данных сохраняются значения времени обработки для коммуникации между мастер-станцией и локальными станциями и удаленными станциями ввода-вывода.</p> <p>За один цикл коммуникации все установленные в параметрах данные связи, например, данные входов (X), выходов (Y), маркеров связи (B) или регистров связи (W), обмениваются с соответствующими станциями сети.</p> <p>Цикл коммуникации представляет собой интервал, за который происходит полный обмен данными со всеми подключенными подчиненными станциями, независимо от времени цикла программы.</p>
D9208*		Минимальное время цикла	
D9209*		Фактическое время цикла	
D9210*	Счетчик повторений	Суммарное количество повторений	После ошибки передачи количество повторных попыток сохраняется в D9210. Счетчик останавливается на максимальном значении FFFFh. Содержимое регистра сбрасывается путем сброса центрального процессора.
D9211*	Счетчик смены линии	Суммарное количество смен	После каждой смены между прямой и обратной линией значение в D9211 повышается на "1".

Специальные регистры связи в мастер-станции

Адрес	Название	Значение	Описание																																																																																					
D9212	Рабочее состояние локальной станции	Состояние станций от 1 до 16	<p>В этих регистрах данных сохраняются адреса локальных станций, находящихся в состоянии PAUSE или STOP.</p> <table border="1"> <tr><td>D9212</td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr> <tr><td>D9212</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr> <tr><td>D9213</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr> <tr><td>D9214*</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr> <tr><td>D9215</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr> </table> <p>Номера станций</p> <p>Если станция находится в состоянии PAUSE или STOP, соответствующий бит устанавливается на „1“. Например, если станция № 7 находится в состоянии STOP, бит 6 в регистре D9212 устанавливается на "1". В этом случае регистр D9212 считается как 64 (40<sub>n</sub>).</p>	D9212	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9212	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	D9213	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	D9214*	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	D9215	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49
D9212		b15		b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																						
D9212		16		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1																																																																						
D9213		32		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17																																																																						
D9214*	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33																																																																								
D9215	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49																																																																								
D9213	Состояние станций от 17 до 32																																																																																							
D9214*	Состояние станций от 33 до 48																																																																																							
D9215*	Состояние станций от 49 до 64																																																																																							
D9216	Распознавание ошибок в локальной станции	Состояние станций от 1 до 16	<p>В этих регистрах данных сохраняются адреса локальных станций, имеющих ошибки.</p> <table border="1"> <tr><td>D9216</td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr> <tr><td>D9216</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr> <tr><td>D9217</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr> <tr><td>D9218*</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr> <tr><td>D9219*</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr> </table> <p>Номера станций</p> <p>Как только в локальной станции распознается ошибка, соответствующий бит устанавливается на „1“. Например, если имеется ошибка в станциях 6 и 12, биты 5 и 11 в регистре D9216 устанавливаются на „1“. В этом случае регистр D9218 считается как 2082 (820<sub>n</sub>).</p>	D9216	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9216	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	D9217	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	D9218*	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	D9219*	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49
D9216		b15		b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																						
D9216		16		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1																																																																						
D9217		32		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17																																																																						
D9218*	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33																																																																								
D9219*	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49																																																																								
D9217	Состояние станций от 17 до 32																																																																																							
D9218*	Состояние станций от 33 до 48																																																																																							
D9219*	Состояние станций от 49 до 64																																																																																							
D9220	Ошибка параметра локальной станции или ошибочное сопоставление входов и выходов удаленной станции ввода-вывода	Состояние станций от 1 до 16	<p>В этих регистрах данных сохраняются адреса локальных станций, данные которых отличаются от данных параметров. Кроме того, сохраняются адреса удаленных станций ввода-вывода с ошибочным присвоением входов и выходов.</p> <table border="1"> <tr><td>D9220</td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr> <tr><td>D9220</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr> <tr><td>D9221</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr> <tr><td>D9222*</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr> <tr><td>D9223*</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr> </table> <p>Номера станций</p> <p>При обнаружении ошибки соответствующий бит устанавливается на „1“. Например, если в локальной станции 5 и в удаленной станции ввода-вывода № 14 имеется ошибка, биты 4 и 13 в регистре D9220 устанавливаются на „1“. В этом случае содержимое регистра D9220 считается как 8208 (2010<sub>n</sub>).</p>	D9220	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9220	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	D9221	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	D9222*	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	D9223*	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49
D9220		b15		b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																						
D9220		16		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1																																																																						
D9221		32		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17																																																																						
D9222*	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33																																																																								
D9223*	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49																																																																								
D9221	Состояние станций от 17 до 32																																																																																							
D9222*	Состояние станций от 33 до 48																																																																																							
D9223*	Состояние станций от 49 до 64																																																																																							
D9224	Инициализирующая передача в локальные станции или удаленные станции ввода-вывода	Состояние станций от 1 до 16	<p>В этих регистрах данных сохраняются адреса локальных и удаленных станций ввода-вывода, инициализируемых со стороны соответствующей мастер-станции.</p> <table border="1"> <tr><td>D9224</td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr> <tr><td>D9224</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr> <tr><td>D9225</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td></tr> <tr><td>D9226*</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td></tr> <tr><td>D9227*</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td></tr> </table> <p>Номера станций</p> <p>Бит станции, получающей данные инициализации, устанавливается на „1“. Например, если в процессе инициализации находится станция № 45, бит 23 в регистре D9226 устанавливается на „1“. В этом случае значение регистра D9226 считается как 4096 (1000<sub>n</sub>).</p>	D9224	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9224	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	D9225	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	D9226*	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	D9227*	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49
D9224		b15		b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																						
D9224		16		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1																																																																						
D9225		32		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17																																																																						
D9226*	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33																																																																								
D9227*	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49																																																																								
D9225	Состояние станций от 17 до 32																																																																																							
D9226*	Состояние станций от 33 до 48																																																																																							
D9227*	Состояние станций от 49 до 64																																																																																							



Специальные регистры связи в мастер-станции

Адрес	Название	Значение	Описание																																																																																																																																																																																																																																																																																																	
D9228	Ошибка в удаленной станции ввода-вывода или локальной станции	Состояние станций от 1 до 16	<p>В этих регистрах помечаются все станции, работающие с ошибками.</p> <table border="1"> <tr> <td></td> <td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>D9228</td> <td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td> </tr> <tr> <td>D9229</td> <td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td> </tr> <tr> <td>D9230</td> <td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td> </tr> <tr> <td>D9231</td> <td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td> </tr> </table> <p>Номер удаленной станции ввода-вывода или локальной станции                      При ошибке присвоенный данной станции бит устанавливается на „1“. Например, если имеется ошибка в локальной станции 36 и удаленной станции ввода-вывода 14, в регистре D9228 биты 2 и 13 устанавливаются на „1“. В этом случае значение регистра D9228 считывается как 8196 (2004h).</p>		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9228	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	D9229	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	D9230	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	D9231	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49																																																																																																																																																																																																												
		b15		b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																																																																																																																																																																																																																																		
D9228		16		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1																																																																																																																																																																																																																																																																																		
D9229		32		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17																																																																																																																																																																																																																																																																																		
D9230	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33																																																																																																																																																																																																																																																																																				
D9231	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49																																																																																																																																																																																																																																																																																				
D9229	Состояние станций от 17 до 32																																																																																																																																																																																																																																																																																																			
D9230*	Состояние станций от 33 до 48																																																																																																																																																																																																																																																																																																			
D9231*	Состояние станций от 49 до 64																																																																																																																																																																																																																																																																																																			
D9232	Ошибка петли локальной или удаленной станции ввода-вывода  Примечание: кроме MELSECNET/B	Состояние станций от 1 до 8	<p>В этих регистрах данных помечаются адреса станций, у которых распознана ошибка в прямой или обратной петле.</p> <table border="1"> <tr> <td></td> <td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>D9232</td> <td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td> </tr> <tr> <td></td> <td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>D9233</td> <td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td> </tr> <tr> <td></td> <td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>D9234</td> <td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td> </tr> <tr> <td></td> <td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>D9235</td> <td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td> </tr> <tr> <td></td> <td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>D9136</td> <td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td> </tr> <tr> <td></td> <td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>D9237</td> <td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td> </tr> <tr> <td></td> <td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>D9238</td> <td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td> </tr> <tr> <td></td> <td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>D9239</td> <td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td><td>R</td><td>V</td> </tr> <tr> <td></td> <td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table> <p>Номера станций</p> <p>В таблице прямая петля обозначена буквой "V", а обратная петля – буквой "R". При обнаружении ошибки соответствующий бит устанавливается на „1“. Например, если имеется ошибка в прямой петле станции 5, бит 8 регистра D9232 устанавливается на „1“. В этом случае значение регистра D9232 считывается как 256 (100h).</p>		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9232	R	V	R	V	R	V	R	V	R	V	R	V	R	V	R	V		8	7	6	5	4	3	2	1									D9233	R	V	R	V	R	V	R	V	R	V	R	V	R	V	R	V		16	15	14	13	12	11	10	9									D9234	R	V	R	V	R	V	R	V	R	V	R	V	R	V	R	V		24	23	22	21	20	19	18	17									D9235	R	V	R	V	R	V	R	V	R	V	R	V	R	V	R	V		32	31	30	29	28	27	26	25									D9136	R	V	R	V	R	V	R	V	R	V	R	V	R	V	R	V		40	39	38	37	36	35	34	33									D9237	R	V	R	V	R	V	R	V	R	V	R	V	R	V	R	V		48	47	46	45	44	43	42	41									D9238	R	V	R	V	R	V	R	V	R	V	R	V	R	V	R	V		56	55	54	53	52	51	50	49									D9239	R	V	R	V	R	V	R	V	R	V	R	V	R	V	R	V		64	63	62	61	60	59	58	57								
		b15		b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																																																																																																																																																																																																																																		
D9232		R		V	R	V	R	V	R	V	R	V	R	V	R	V	R	V																																																																																																																																																																																																																																																																																		
		8		7	6	5	4	3	2	1																																																																																																																																																																																																																																																																																										
D9233		R		V	R	V	R	V	R	V	R	V	R	V	R	V	R	V																																																																																																																																																																																																																																																																																		
		16		15	14	13	12	11	10	9																																																																																																																																																																																																																																																																																										
D9234		R		V	R	V	R	V	R	V	R	V	R	V	R	V	R	V																																																																																																																																																																																																																																																																																		
		24		23	22	21	20	19	18	17																																																																																																																																																																																																																																																																																										
D9235		R		V	R	V	R	V	R	V	R	V	R	V	R	V	R	V																																																																																																																																																																																																																																																																																		
	32	31	30	29	28	27	26	25																																																																																																																																																																																																																																																																																												
D9136	R	V	R	V	R	V	R	V	R	V	R	V	R	V	R	V																																																																																																																																																																																																																																																																																				
	40	39	38	37	36	35	34	33																																																																																																																																																																																																																																																																																												
D9237	R	V	R	V	R	V	R	V	R	V	R	V	R	V	R	V																																																																																																																																																																																																																																																																																				
	48	47	46	45	44	43	42	41																																																																																																																																																																																																																																																																																												
D9238	R	V	R	V	R	V	R	V	R	V	R	V	R	V	R	V																																																																																																																																																																																																																																																																																				
	56	55	54	53	52	51	50	49																																																																																																																																																																																																																																																																																												
D9239	R	V	R	V	R	V	R	V	R	V	R	V	R	V	R	V																																																																																																																																																																																																																																																																																				
	64	63	62	61	60	59	58	57																																																																																																																																																																																																																																																																																												
D9233	Состояние станций от 9 до 16																																																																																																																																																																																																																																																																																																			
D9234	Состояние станций от 17 до 24																																																																																																																																																																																																																																																																																																			
D9235	Состояние станций от 25 до 32																																																																																																																																																																																																																																																																																																			
D9236	Состояние станций от 33 до 40																																																																																																																																																																																																																																																																																																			
D9237	Состояние станций от 41 до 48																																																																																																																																																																																																																																																																																																			
D9238	Состояние станций от 49 до 56																																																																																																																																																																																																																																																																																																			
D9239	Состояние станций от 57 до 64																																																																																																																																																																																																																																																																																																			
D9240	Счетчик ошибок передачи	Счетчик ошибок передачи	Как только распознается одна из ошибок передачи CRC, OVER или AB.IF, значение регистра D9240 повышается на „1“. Счетчик останавливается на максимальном значении FFFFh. Содержимое регистра сбрасывается путем сброса центрального процессора.																																																																																																																																																																																																																																																																																																	
D9241*	Тип связи локальной станции  (см. также D9202, D9203)	Состояние станций от 33 до 48	<p>В этих регистрах данных сохраняется информация о совместимости подчиненных станций (slave) с сетями MELSECNET и MESECNET II.</p> <p>Если станция совместима с MELSECNET II, в сопоставленный этой станции бит специального регистра записывается „1“. Если станция совместима с MELSECNET, бит устанавливается на „0“.</p>																																																																																																																																																																																																																																																																																																	
D9242*		Состояние станций от 49 до 64																																																																																																																																																																																																																																																																																																		
D9243	Контроль номера станции	Номер станции (от 0 до 64)	Этот регистр данных служит для контроля собственного адреса локальной станции.																																																																																																																																																																																																																																																																																																	
D9244	Количество подключенных подчиненных станций (slave)	Общее количество подчиненных станций (slave)	В этом регистре данных сохраняется суммарное количество подчиненных станций, находящихся на одной линии.																																																																																																																																																																																																																																																																																																	
D9245	Счетчик ошибок передачи	Суммарное количество ошибок передачи	Как только распознается одна из ошибок передачи CRC, OVER или AB.IF, значение в D9245 повышается на „1“. Счетчик останавливается на максимальном значении FFFFh. Содержимое регистра сбрасывается путем сброса центрального процессора.																																																																																																																																																																																																																																																																																																	

Специальные регистры связи в мастер-станции

Адрес	Название	Значение	Описание																																																																																					
D9248	Рабочее состояние локальной станции	Состояние станций от 1 до 16	<p>В этих регистрах данных помечены все локальные станции, находящиеся в состоянии PAUSE или STOP.</p> <table border="1"> <tr> <td></td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>D9196</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td> </tr> <tr> <td>D9197</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td> </tr> <tr> <td>D9198</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td> </tr> <tr> <td>D9199</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td> </tr> </table> <p>Номера станций</p> <p>Если локальная станция находится в состоянии PAUSE или STOP, соответствующий бит устанавливается на "1". Например, при переключении станций 7 и 15 в состояние STOP биты 6 и 14 в D9248 устанавливаются на „1“. В этом случае значение регистра D9248 считается как 16448 (4040<sub>н</sub>).</p>		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9196	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	D9197	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	D9198	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	D9199	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49
		b15		b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																						
D9196		16		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1																																																																						
D9197		32		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17																																																																						
D9198	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33																																																																								
D9199	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49																																																																								
D9249	Состояние станций от 17 до 32																																																																																							
D9250*	Состояние станций от 33 до 48																																																																																							
D9251*	Состояние станций от 49 до 64																																																																																							
D9252	Ошибка в локальной станции	Состояние станций от 1 до 16	<p>В этих регистрах данных отмечается локальная станция вне хост-станции, в которой имеется неисправность.</p> <table border="1"> <tr> <td></td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>D9196</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td> </tr> <tr> <td>D9197</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td> </tr> <tr> <td>D9198</td><td>48</td><td>47</td><td>46</td><td>45</td><td>44</td><td>43</td><td>42</td><td>41</td><td>40</td><td>39</td><td>38</td><td>37</td><td>36</td><td>35</td><td>34</td><td>33</td> </tr> <tr> <td>D9199</td><td>64</td><td>63</td><td>62</td><td>61</td><td>60</td><td>59</td><td>58</td><td>57</td><td>56</td><td>55</td><td>54</td><td>53</td><td>52</td><td>51</td><td>50</td><td>49</td> </tr> </table> <p>Номера станций</p> <p>Как только в локальной станции обнаруживается ошибка, соответствующий бит устанавливается на „1“. Например, если имеется ошибка в станции 6, бит 5 в D9252 устанавливается на „1“. В этом случае значение регистра D9252 считается как 2048 (800<sub>н</sub>).</p>		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	D9196	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	D9197	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	D9198	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	D9199	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49
		b15		b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																																																						
D9196		16		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1																																																																						
D9197		32		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17																																																																						
D9198	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33																																																																								
D9199	64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49																																																																								
D9253	Состояние станций от 17 до 32																																																																																							
D9254*	Состояние станций от 33 до 48																																																																																							
D9255*	Состояние станций от 49 до 64																																																																																							

\* Специальные регистры, обозначенные этой звездочкой, не могут применяться в сети MELSECNET/B.

# Указатель ключевых слов

<b>A</b>		<b>C</b>	
ACOS, ACOSP . . . . .	7-365	CALL, CALLP . . . . .	7-134
ADRSET, ADRSETP . . . . .	7-493	CGMODE . . . . .	10-4
AnA, AnAS и AnU . . . . .	3-38	CHG. . . . .	7-150
ANB, ORB . . . . .	5-11	CHK . . . . .	5-48
ANDP, ANDF . . . . .	5-8	CHK (серия "A") . . . . .	7-232
AND, ANI . . . . .	5-4	CHKCIR, CHKEND . . . . .	7-240
ASC (серия "A") . . . . .	7-324	CHKST, CHK (серии "Q" и "System Q") . . . . .	7-224
ASC, ASCP (серии "Q" и "System Q") . . . . .	7-320	CJ . . . . .	6-150
ASIN, ASINP . . . . .	7-362	CLOSE . . . . .	11-57
ATAN, ATANP . . . . .	7-368	CML, CMLP . . . . .	6-129
<b>B</b>		COM . . . . .	6-174
B +, B + P . . . . .	6-43	COMRD, COMRDP . . . . .	7-285
B -, B - P . . . . .	6-43	COS, COSP . . . . .	7-356
B x, B x P . . . . .	6-53	<b>D</b>	
B/, B/P . . . . .	6-53	DABCD, DABCDP . . . . .	7-280
BACOS, BACOSP . . . . .	7-404	DABIN, DABINP . . . . .	7-271
BAND, BANDP . . . . .	7-416	DAND, DANDP . . . . .	7-4
BASIN, BASINP . . . . .	7-401	DATERD, DATERDP . . . . .	7-435
BATAN, BATANP . . . . .	7-407	DATEWR, DATEWRP . . . . .	7-440
BBLKRD, BBLKRDP . . . . .	11-28	DATE-, DATE-P . . . . .	7-450
BBLKWR, BBLKWRP . . . . .	11-31	DATE+, DATE+P . . . . .	7-445
BCD, BCDP . . . . .	6-83	DB+, DB+P . . . . .	6-48
BCDDA, BCDDAP . . . . .	7-266	DB-, DB-P . . . . .	6-48
BCOS, BCOSP . . . . .	7-395	DBx, DBxP . . . . .	6-56
BDSQR, BDSQRP . . . . .	7-388	DB/, DB/P . . . . .	6-56
BIN, BINP . . . . .	6-86	DBAND, DBANDP . . . . .	7-416
BINDA, BINDAP . . . . .	7-256	DBCDDA, DBCDDAP . . . . .	7-266
BINHA, BINHAP . . . . .	7-261	DBC, DBCDP . . . . .	6-83
BK-, BK-P . . . . .	6-68	DBINDA, DBINDAP . . . . .	7-256
BK+, BK+P . . . . .	6-68	DBINHA, DBINHAP . . . . .	7-261
BKAND, BKANDP . . . . .	7-11	DBIN, DBINP . . . . .	6-86
BKBCD, BKBCDP . . . . .	6-113	DBL, DBLP . . . . .	6-98
BKBIN, BKBINP . . . . .	6-116	DCML, DCMLP . . . . .	6-129
BKCMP, BKCMPP . . . . .	6-20	DDABCD, DDABCDP . . . . .	7-280
BKOR, BKORP . . . . .	7-20	DDABIN, DDABINP . . . . .	7-271
BKRST, BKRSTP . . . . .	7-73	DDEC, DDECP . . . . .	6-78
BKXNR, BKXNRP . . . . .	7-39	DECO, DECOP . . . . .	7-88
BKXOR, BKXORP . . . . .	7-29	DEC, DECP . . . . .	6-75
BMOV, BMOV P . . . . .	6-134	DEG, DEGP . . . . .	7-374
BREAK, BREAKP . . . . .	7-131	DELTA, DELTAP . . . . .	5-50
BRST, BRSTP . . . . .	7-66	DFLT, DFLTP . . . . .	6-90
BSET, BSETP . . . . .	7-66	DFRO, DFROP . . . . .	7-190
BSFL, BSFLP . . . . .	7-59	DGBIN, DGBINP . . . . .	6-105
BSFR, BSFRP . . . . .	7-59	DGRY, DGRYP . . . . .	6-102
BSIN, BSINP . . . . .	7-392	DHABIN, DHABINP . . . . .	7-276
BSQR, BSQRP . . . . .	7-388	DI . . . . .	6-158
BTAN, BTANP . . . . .	7-398	DINC, DINCP . . . . .	6-78
BTOW, BTOWP . . . . .	7-107	DINT, DINTP . . . . .	6-94
BUFRCV . . . . .	11-35	DIS, DISP . . . . .	7-96
BUFRCVS (интерфейсные модули) . . . . .	11-3	DLIMIT, DLIMITP . . . . .	7-412
BUFRCVS (модуль Ethernet) . . . . .	11-40	DMAX, DMAXP . . . . .	7-113
BUFSND . . . . .	11-43	DMIN, DMINP . . . . .	7-116
BXCH, BXCHP . . . . .	6-143	DMOV, DMOV P . . . . .	6-120

DNEG, DNEGP	6-108
DOR, DORP	7-14
DRCL, DRCLP	7-52
DRCR, DRCRP	7-49
DROL, DROLP	7-52
DROR, DRORP	7-49
DSER, DSERP	7-79
DSFL, DSFLP	7-62
DSFR, DSFRP	7-62
DSORT, DSORTP	7-119
DSTR, DSTRP	7-292
DSUM, DSUMP	7-85
DTEST, DTESTP	7-69
DTO, DTO P	7-194
DUTY	7-482
DVAL, DVALP	7-299
DWSUM, DWSUMP	7-125
DXCH, DXCHP	6-140
DXNR, DXNRP	7-32
DXOR, DXORP	7-23
DZONE, DZONEP	7-420

**E**

E+, E+P	6-60
E-, E-P	6-60
Ex, ExP	6-65
E/, E/P	6-65
ECALL, ECALLP	7-143
EFCALL, EFCALLP	7-146
EGP, EGF	5-21
EI	6-158
EI, DI	6-177
EMOD, EMODP	7-344
EMOV, EMOVP	6-123
ENCO, ENCO P	7-90
END	5-64
ENEG, ENEGP	6-111
EREXP, EREXP P	7-348
EROMW R, EROMW RP	7-503
ERRCLR	11-62
ERRRD	11-68
ESTR, ESTRP	7-305
Ethernet	
новая инициализация	11-75
стирание индикации ошибки	11-64
считывание кода ошибки	11-70
EVAL, EVALP	7-314
EXP, EXPP	7-380

**F**

FCALL, FCALLP	7-139
FDEL, FDELP	7-183
FEND	5-61
FF	5-46
FIFR, FIFRP	7-175
FIFW, FIFWP	7-171
FINS, FINSP	7-183
FLT, FLTP	6-90
FMOV, FMOVP	6-137
FOR, NEXT	7-128
FPOP, FPOPP	7-179
FREAD	9-20
FROM, DFRO	7-190
FWRITE	9-9

**G**

GBIN, GBINP	6-105
GETE, GETEP	11-6
GOEND	6-155
GRY, GRYP	6-102

**H**

HABIN, HABINP	7-276
HEX, HEXP	7-326
HOUR, HOURP	7-455

**I**

IMASK	6-158
INC, INCP	6-75
INSTR, INSTRP	7-340
INT, INTP	6-94
INV	5-17
IRET	6-165
IXDEV, IXSET	7-167
IX, IXEND	7-162

**J**

JMP	6-150
-----	-------

**K**

KEY	7-494
-----	-------

**L**

LD	6-5
LD, LDI	5-4
LD=	6-5
LD>	6-5
LD>=	6-5
LDP, LDF	5-8
LED	7-211
LEDA, LEDB	7-217
LEDC	7-214
LEDR	7-219
LEFT, LEFTP	7-330
LEN, LENP	7-289
LIMIT, LIMITP	7-412
LOG, LOGP	7-383
LRDP	8-86
LWTP	8-90

**M**

MAX, MAXP	7-113
MC, MCR	5-55
MELSECNET	
попарное обобщение станций	11-81
MEP, MEF	5-19
MIDR, MIDRP	7-334
MIDW, MIDWP	7-334
MIN, MINP	7-116
MOV, MOVP	6-120
MPS, MRD, MPP	5-14
MSG	7-462
MTR	6-206
M_REAL_TO_REAL	3-18
M_REAL_TO_REAL_E	3-18

<b>N</b>	
NDIS, NDISP . . . . .	7-102
NEG, NEGP. . . . .	6-108
NOP . . . . .	5-70
NUNI, NUNIP . . . . .	7-102
<b>O</b>	
OPEN . . . . .	11-48
OR, ORI . . . . .	5-4
ORP, ORF. . . . .	5-8
OUT . . . . .	5-23
OUT C. . . . .	5-28
OUT F. . . . .	5-31
OUT T, OUTH T . . . . .	5-25
<b>P</b>	
PAIRSET . . . . .	11-80
PKEY. . . . .	7-465
PLOADP . . . . .	9-33
PLOW, PLOWP . . . . .	7-475
PLS, PLF . . . . .	5-42
PLSY. . . . .	6-202
POFF, POFFP . . . . .	7-471
PR . . . . .	7-200
PRC . . . . .	7-206
PRR, PRRP . . . . .	11-19
PSCAN, PSCANP . . . . .	7-473
PSTOP, PSTOPP . . . . .	7-469
PSWAPP . . . . .	9-38
PTRA, PTRAR . . . . .	7-251
PTRAEXE, PTRAEXEP . . . . .	7-251
PUNLOADP . . . . .	9-36
PUTE, PUTEP . . . . .	11-12
PWM. . . . .	6-204
<b>Q</b>	
QCDSET, QCDSETP . . . . .	7-431
QDRSET, QDRSETP . . . . .	7-428
Q00CPU . . . . .	1-2
Q00JCPU . . . . .	1-2
Q01CPU . . . . .	1-2
Q02CPU . . . . .	1-2
Q02HCPU . . . . .	1-2
Q06HCPU . . . . .	1-2
Q12HCPU . . . . .	1-2
Q4ARCPU . . . . .	
передача данных. . . . .	10-6
Q4ARCPU . . . . .	
выбор режима . . . . .	10-2
<b>R</b>	
RAD, RADP. . . . .	7-371
RAMP . . . . .	6-198
RBMOV, RBMOV P . . . . .	9-41
RCL, RCLP . . . . .	7-46
RCR, RCRP . . . . .	7-43
READ . . . . .	8-12
обзор операндов управляющих данных . . . . .	8-14
REAL_TO_M_REAL . . . . .	3-18
REAL_TO_M_REAL_E. . . . .	3-18
RECV . . . . .	8-48
REQ . . . . .	8-53
RET . . . . .	7-137
RFRP . . . . .	8-94
RFS, RFSP . . . . .	6-168
RIFR (серия "A"). . . . .	11-169
RIFR (QnA, System Q) . . . . .	11-173
RIGHT, RIGHTP. . . . .	7-330
RIRCV (серия "A"). . . . .	11-137
RIRCV (QnA, System Q) . . . . .	11-143
RIRD (серия "A") . . . . .	11-109
RIRD (QnA, "System Q"). . . . .	11-115
RISEND (серия "A"). . . . .	11-149
RISEND (QnA, System Q) . . . . .	11-155
RITO (серия "A") . . . . .	11-161
RITO (QnA, System Q) . . . . .	11-165
RIWT (серия "A") . . . . .	11-123
RIWT (QnA, System Q). . . . .	11-129
RLPA (серия "A") . . . . .	11-84
RLPASET, RLPASET_P . . . . .	11-90
RND, RNDP . . . . .	7-386
ROL, ROLP. . . . .	7-46
ROR, RORP . . . . .	7-43
ROTC . . . . .	6-193
RRPA (серия "A") . . . . .	11-102
RSET, RSETP . . . . .	7-425
RST . . . . .	5-36
RST F . . . . .	5-39
RTOP . . . . .	8-100
RTREAD . . . . .	8-107
RTWRITE. . . . .	8-109
<b>S</b>	
SCJ . . . . .	6-150
SECOND, SECONDP . . . . .	7-455
SEG . . . . .	6-170
SEG, SEGP. . . . .	7-92
SEND . . . . .	8-40
SER, SERP . . . . .	7-79
SET . . . . .	5-34
SET F . . . . .	5-39
SFL, SFLP . . . . .	7-56
SFR, SFRP . . . . .	7-56
SFT, SFTP . . . . .	5-52
SIN, SINP. . . . .	7-353
SLT, SLTR . . . . .	7-245
SORT, SORTP . . . . .	7-119
SPD . . . . .	6-200
SPREF. . . . .	10-11
SQR, SQRP . . . . .	7-377
SREAD . . . . .	8-19
состояние выполнения . . . . .	8-23
состояние после завершения . . . . .	8-23
флаг коммуникационного канала . . . . .	8-23
хост-станция . . . . .	8-23
SRND, SRNDP . . . . .	7-386
STC, CLC. . . . .	7-480
STMODE. . . . .	10-2
STMR, STMRH . . . . .	6-189
S.TO, SP.TO . . . . .	9-46
STOP . . . . .	5-67
STRA, STRAR . . . . .	7-248

STR, STRP	7-292
SUB, SUBP	7-159
SUM, SUMP	7-85
SWAP, SWAPP	6-146
SWRITE	8-33
System Q	
многопроцессорные модули центрального процессора	1-2
однопроцессорные модули центрального процессора	1-2

**T**

TAN, TANP	7-359
TEST, TESTP	7-69
TO, DTO	7-194
TRACE, TRACER	9-7
TRUCK	10-6
TTMR	6-187

**U**

UDCNT1	6-181
UDCNT2	6-18
UINI	11-73
UNI, UNIP	7-99
UNIRD, UNIRDP	9-2

**V**

VAL, VALP	7-299
-----------	-------

**W**

WAND, WANDP	7-4
WDT, WDTP	7-478
WOR, WORP	7-14
WORD, WORDP	6-100
WRITE	8-26
WSUM, WSUMP	7-123
WTOB, WTOBP	7-107
WXNR	7-33
WXNR, WXNRP	7-32
WXOR, WXORP	7-23

**X**

XCH, XCHP	6-140
-----------	-------

**Z**

ZCOM	8-7
ZNFR	8-65
ZNRD	8-78
ZNTO	8-71
ZNWR	8-82
ZONE, ZONEP	7-420
ZPOP, ZPOPP	7-500
ZPUSH, ZPUSHP	7-500
ZRRDB, ZRRDBP	7-485
ZRWRB, ZRWRBP	7-489

**A**

Адрес перехода	6-154
Адрес указателя	6-154
Адреса контактов	7-242
Адреса операндов в случае процессоров	
Адресация массивов	3-20
Адресация массивов и регистров	3-19
Арифметические операции	6-25
над двоично-десятичными данными	6-27

**Б**

Базовый набор команд	5-1
Библиотека изготовителя (Manufacturer_Lib)	3-4
Битовые блоки	3-14
Битовые блоки в случае данных типа "двойное слово"	3-15
Битовые блоки (указание для d)	3-13
Битовые операнды и битовые данные	3-11
Битовые операнды и словные данные	3-12
Блочная обработка битовых данных	3-12
Буквенно-числовые строковые величины	7-325

**В**

Ввод данных в общую память	9-47
Ввод с клавиатуры	
на периферийных устройствах	7-466
числовых значений	7-495
Возврат в основную программу	7-138
Возрастающий/убывающий счетчик	
двухфазный	6-185
однофазный	6-182
Вращение данных	
влево (16 бит)	7-47
влево (32 бита)	7-53
вправо (16 битов)	7-44
вправо (32 бита)	7-50
Вставление определенных блоков данных	7-185
Вход EN	3-35
Входные команды	5-4
Входные условия команд	3-34
Выборочная трассировка	7-223
сброс	7-249
установка	7-249
Вывод	
комментария	7-207
маркеров ошибки	5-32
на светодиодный дисплей (данных в формате ASCII)	7-212
на светодиодный дисплей (комментариев)	7-215
на светодиодный дисплей (строковой величины ASCII)	7-218
сообщений на периферийные устройства	7-463
строковой величины ASCII	7-201
Вывод по фронту сигнала	5-43
Вызов подпрограммы (Call)	7-135
Вызов подпрограммы (ECALL)	7-144
Вызов программы микрокомпьютера	7-160
Выход ENO	3-35
Выход таймера	6-190
Вычисление арккосинуса чисел с плавающей запятой	7-366

Вычисление арксинуса	
двоично-десятичных данных . . . . .	7-402
чисел с плавающей запятой . . . . .	7-363
Вычисление арктангенса	
двоично-десятичных данных . . . . .	7-408
чисел с плавающей запятой . . . . .	7-369
Вычисление квадратного корня . . . . .	7-378
4-значных двоично-десятичных данных . . . . .	7-389
Вычисление косинуса	
двоично-десятичных данных . . . . .	7-396
чисел с плавающей запятой . . . . .	7-357
Вычисление натурального логарифма. . . . .	7-384
Вычисление синуса	
двоично-десятичных данных . . . . .	7-393
чисел с плавающей запятой . . . . .	7-354
Вычисление тангенса	
двоично-десятичных данных . . . . .	7-399
чисел с плавающей запятой . . . . .	7-360
Вычисление экспоненты . . . . .	7-381
Вчитание данных времени . . . . .	7-451
Выявление и устранение ошибок . . . . .	7-223

**Г**

Головной адрес. . . . .	7-197
Группирование 16-битных данных . . . . .	7-100

**Д**

Данные для 7-сегментных дисплеев . . . . .	7-94
Данные запроса REQ . . . . .	8-57
Данные ответа REQ . . . . .	8-58
Данные типа REAL . . . . .	7-310
Декодировка	
данных с 8 на 256 бит. . . . .	7-89
для 7-сегментных дисплеев . . . . .	7-93
Декодировка данных. . . . .	7-89
Десятичное. . . . .	7-316
модуля . . . . .	2-63
Десятичное представление	
числа с плавающей запятой . . . . .	7-315
числа типа REAL . . . . .	7-307
Диагностика неисправностей. . . . .	7-237
Диагностика ошибок	
с помощью специальных маркеров. . . . .	A-45
Диаграмма изменения сигналов. . . . .	7-205

**З**

Загрузить (замыкающий контакт) . . . . .	5-5
Загрузить инверсно (размыкающий контакт) . . . . .	5-5
Загрузка	
программы с карты памяти . . . . .	9-34
Заголовок программы на языке IL для MELSEC . . . . .	3-7
Задание количества циклов выполнения. . . . .	7-483
Задание смещения для входа . . . . .	7-411
Задание смещения для выхода . . . . .	7-411
Закрывание связи (Ethernet). . . . .	11-59
Запись	
в буферную память . . . . .	7-195
в регистр файлов EEPROM . . . . .	7-504
в индексный перечень. . . . .	7-168
в словные операнды других станций (SWRITE) . . . . .	8-37
в специальные модули удаленной станции	

ввода-вывода . . . . .	8-73
данных в другие станции. . . . .	8-83
данных в другие станции (WRITE) . . . . .	8-30
данных в локальную станцию . . . . .	8-91
данных в удаленную станцию . . . . .	8-102
данных времени . . . . .	7-441
информации маршрутизации . . . . .	8-110
1-словных данных в специальные модули . . . . .	7-195
2-словных данных в специальные модули . . . . .	7-195
Запись данных	
в другую станцию (CC-Link) . . . . .	11-126
в интеллектуальную станцию CC-Link . . . . .	11-151
в модуль Profibus . . . . .	11-32
в станцию CC-Link . . . . .	11-133
в фиксированные буферы (Ethernet). . . . .	11-45
Запись данных в перечень. . . . .	7-172
Запись параметров маршрутизация (RTWRITE) . . . . .	8-110
Запрос данных от другой станции . . . . .	8-60

**И**

Иерархия типов данных ANY . . . . .	3-9
Изменение арифметического знака	
чисел с плавающей запятой. . . . .	6-112
Изменение коммутационного состояния на	
противоположное . . . . .	5-49
Измерение длины строковых величин . . . . .	7-290
Инвертирование битового выходного операнда . . . . .	5-47
Инвертирование данных	
16 битов . . . . .	6-130
32 бита . . . . .	6-130
Индексация . . . . .	3-24
в процессорах AnA, AnAS и AnU . . . . .	3-28
в процессорах QnA . . . . .	3-26
Индексированная адресация. . . . .	7-163
Информация для диагностики . . . . .	A-77
Информация цикла. . . . .	A-53
Информация цикла программы . . . . .	A-97
Исключающее ИЛИ . . . . .	7-24
для 16-битных блоков . . . . .	7-30
Исключающее НЕ ИЛИ. . . . .	7-33
для 16-битных блоков данных . . . . .	7-40
Источник данных (s) в команде . . . . .	3-1
Источники ошибок . . . . .	4-6

**К**

Карты памяти . . . . .	A-54
Катушка таймера . . . . .	6-190
Кодировка	
данных из 256 в 8 бит . . . . .	7-91
Кодировка данных . . . . .	7-91
Количество (n) в команде . . . . .	3-2
Количество шагов программы	
в случае процессоров AnA, AnAS и AnU . . . . .	3-38
Команда	
для импульсов переключения . . . . .	5-51
контроля на ошибки . . . . .	7-233
Команда CHG	
в сочетании с командой PLS . . . . .	7-153
в сочетании с импульсной командой . . . . .	7-154
время таймеров . . . . .	7-156
обработка команд OUT . . . . .	7-156
принцип работы счетчиков . . . . .	7-155

Команда вращения		
влево (16 бит), без флага переноса . . . . .	7-47	
влево (16 бит), с флагом переноса . . . . .	7-47	
влево (32 бита), без флага переноса . . . . .	7-53	
влево (32 бита), с флагом переноса . . . . .	7-53	
вправо (16 бит), с флагом переноса . . . . .	7-44	
вправо (32 бита), без флага переноса . . . . .	7-50	
вправо (32 бита), с флагом переноса . . . . .	7-50	
Команда вращения битов		
вправо (16 бит), без флага переноса . . . . .	7-44	
Команда вывода импульсов . . . . .	6-203	
Команда инвертирования . . . . .	5-18	
Команда позиционирования поворотных столов . . . . .	6-195	
Команда установки		
выходов . . . . .	5-24	
контактов таймера . . . . .	5-26	
Команды		
для выходных контактов . . . . .	5-23	
преобразования массива в начальный адрес . . . . .	3-21	
Команды вращения битов . . . . .	7-42	
Команды для Q4ARCPU . . . . .	10-1	
Команды для вызова программ прерывания . . . . .	6-157	
Команды для многопроцессорного режима . . . . .	9-46	
Команды для переключения блоков		
регистров файлов . . . . .	7-424	
Команды для перечней данных . . . . .	7-170	
Команды для периферийных устройств . . . . .	7-461	
Команды для процессоров MELSEC System Q . . . . .	9-1	
Команды для специальных модулей . . . . .	11-1	
Команды для структурирования программы . . . . .	7-127	
Команды для управления программой . . . . .	7-468	
Команды для часов . . . . .	7-434	
Команды доступа к буферной памяти . . . . .	7-189	
Команды индикации . . . . .	7-198	
Команды конвертирования . . . . .	6-81	
Команды коммуникации . . . . .	8-1	
области записи и считывания . . . . .	8-3	
Команды коммуникации (серия "А") . . . . .	8-77	
Команды контроля данных . . . . .	7-411	
Команды обновления данных . . . . .	6-167, 8-6	
Команды обработки битов . . . . .	7-65	
Команды обработки данных . . . . .	7-77	
Команды обращения с программой для процессоров "System Q" . . . . .	9-33	
Команды передачи . . . . .	6-119	
Команды передачи для процессоров "System Q" . . . . .	9-41	
Команды сдвига . . . . .	7-55	
Команды сдвига (битовые операнды) . . . . .	5-52	
Компилятор . . . . .	3-19	
Конвертирование		
данных в кодировке Грея в 16-битные двоичные данные . . . . .	6-106	
данных в кодировке Грея в 32-битные двоичные данные . . . . .	6-106	
двоично-десятичных данных в ASCII-код . . . . .	7-267	
двоично-десятичных данных (4 знака) в двоичные . . . . .	6-87	
двоично-десятичных данных (8 знаков) в двоичные . . . . .	6-87	
двоичных данных в двоично-десятичные (4-значные) . . . . .	6-84	
двоичных данных в двоично-десятичные (8-значные) . . . . .	6-84	
двоичных данных в строковые величины . . . . .	7-293	
двоичных данных в шестнадцатеричные числа (ASCII) . . . . .	7-262	
двоичных данных (16 бит) в числа с плавающей запятой . . . . .	6-91	
двоичных данных (32 бита) в числа с плавающей запятой . . . . .	6-91	
двоичных чисел в десятичные (ASCII-код) . . . . .	7-257	
десятичных данных в кодировке ASCII в двоично-десятичные данные . . . . .	7-281	
десятичных данных в кодировке ASCII в двоичные данные . . . . .	7-272	
строковых величин в двоичные данные . . . . .	7-300	
строковых величин в десятичные числа с плавающей запятой . . . . .	7-315	
строковых величин в коды ASCII . . . . .	7-325	
шестнадцатеричных данных в кодировке ASCII в двоичные данные . . . . .	7-277	
шестнадцатеричных значений в ASCII-кодировке в двоичные значения . . . . .	7-327	
чисел с плавающей запятой в строковые величины . . . . .	7-306	
чисел с плавающей запятой в 32-битные двоичные данные . . . . .	6-95	
числа с плавающей запятой в 16-битные двоичные данные . . . . .	6-95	
16-битных двоичных данных в код Грея . . . . .	6-103	
16-битных двоичных данных в коды ASCII . . . . .	7-322	
16-битных двоичных данных в 32-битные двоичные данные . . . . .	6-99	
16-битных двоичных данных в 4-значные двоично-десятичные числа . . . . .	6-114	
32-битных двоичных данных в код Грея . . . . .	6-103	
32-битных двоичных данных в 16-битные двоичные данные . . . . .	6-101	
4-значных двоично-десятичных чисел в 16-битные двоичные данные . . . . .	6-117	
Конец		
области программы . . . . .	5-62	
основной программы или подпрограммы . . . . .	5-65	
программы прерывания . . . . .	6-166	
Конец передачи данных . . . . .	8-4	
Константа в команде . . . . .	3-1	
Контакт таймера . . . . .	6-190	
Контроль битов данных		
16 бит . . . . .	7-86	
32 бита . . . . .	7-86	
Конфигурация команд . . . . .	3-1	
Косвенная адресация . . . . .	3-29	
присвоение операнда . . . . .	7-493	
<b>Л</b>		
Линейно изменяющийся сигнал . . . . .	6-199	
Логические команды . . . . .	7-2	
Логическое "ИЛИ" . . . . .	7-15	
Логическое "И" . . . . .	7-6	
Локальная станция . . . . .	A-75	
<b>М</b>		
Мастер-станция (специальные регистры связи) . . . . .	A-124	
Мастер-станция (специальные маркеры связи) . . . . .	A-73	
Мастер-станция и команды коммуникации . . . . .	8-4	
Место назначения (d) в команде . . . . .	3-2	



**Н**

Название команды . . . . .	2-4
Начало операции . . . . .	5-5
Начальная команда для команды СНК . . . . .	7-225
Начальный регистр . . . . .	3-19

**О**

## Обзор

адресов кодов ошибок . . . . .	7-238
арифметические команды . . . . .	2-15
время обработки (определение) . . . . .	A-1
входных команд . . . . .	2-6
выходных команд . . . . .	2-8
значений времени обработки . . . . .	A-2
информации маршрутизации . . . . .	2-62
кодов ошибок (Q00J, Q00, Q01) . . . . .	13-2
кодов ошибок (процессоры серий QnA, Q02, Q02H, Q06H, Q12H и Q25H) . . . . .	13-12
кодов ошибок серии "A" . . . . .	13-38
кодов ошибок серии AnA и AnAS . . . . .	13-42
команд . . . . .	2-4
команд вращения битов . . . . .	2-35
команд диагностики ошибок . . . . .	2-63
команд для вызова программ прерывания . . . . .	2-27
команд для работы с перечнями данных . . . . .	2-43
команд для обработки строковых величин . . . . .	2-48
команд для особых функций . . . . .	2-52
команд для переключения блоков регистров файлов . . . . .	2-56
команд для периферийной аппаратуры . . . . .	2-58
команд для процессоров серии "System Q" . . . . .	2-63
команд для часов . . . . .	2-57
команд индикации . . . . .	2-45
команд конвертирования . . . . .	2-22
команд конца программы . . . . .	2-9
команд контроля данных . . . . .	2-55
команд обработки битов . . . . .	2-37
команд обработки данных . . . . .	2-38
команд обновления данных . . . . .	2-28
команд обновления сетевых данных . . . . .	2-61
команд передачи . . . . .	2-25
команд передачи (System Q) . . . . .	2-65
команд передачи данных . . . . .	2-64
команд передачи программы . . . . .	2-64
команд сдвига . . . . .	2-36
команд сдвига (битовые операнды) . . . . .	2-8
команд структурирования программы . . . . .	2-41
команды обращения к буферной памяти . . . . .	2-44
логических команд . . . . .	2-31
логических операций . . . . .	2-7
маркеров диагностики (для команд) . . . . .	A-55
маркеров диагностики (серия "Q" и "System Q") . . . . .	A-44
операторов сравнения . . . . .	2-10
организующие команды . . . . .	2-9
применимых операндов . . . . .	A-33
программных команд . . . . .	2-58
прочие команды, часть 1 . . . . .	2-29
прочие руководства . . . . .	1-1
прочих команд (STOP, NOP) . . . . .	2-9
прочих команд, часть 2 . . . . .	2-59

регистров диагностики (серии "Q" и "System Q") . . . . .	A-76
режимов обработки ввода-вывода . . . . .	A-35
специализированных команд передачи данных . . . . .	2-61
специальных маркеров . . . . .	A-44
специальных маркеров (серия "A") . . . . .	A-67
специальных маркеров (серия "A") . . . . .	A-75
специальных маркеров и маркеров диагностики . . . . .	A-61
специальных маркеров связи (серия "A") . . . . .	A-73
специальных регистров . . . . .	A-76
специальных регистров (серия "A") . . . . .	A-114
специальных регистров (только серия "A") . . . . .	A-114
специальных регистров и регистров диагностики . . . . .	A-108
специальных регистров связи (серия "A") . . . . .	A-124
сравнение команд для серий "A" и "Q"/ "System Q" . . . . .	A-41
сравнение команд процессоров QnA и System Q . . . . .	A-42
типов данных . . . . .	A-35
типов команд . . . . .	2-1
типов контроллеров . . . . .	1-2
типов центральных процессоров . . . . .	1-2
функций счетчиков . . . . .	A-39
функций таймеров . . . . .	A-36
функциональные блоки таймеров . . . . .	A-38
функциональных блоков счетчиков . . . . .	A-40
Обзор операндов управляющих данных	
RECV . . . . .	8-49
REQ . . . . .	8-55
SEND . . . . .	8-41
SREAD . . . . .	8-21
SWRITE . . . . .	8-35
WRITE . . . . .	8-28
ZNFR . . . . .	8-66
ZNT0 . . . . .	8-72
Обработка	
битовых данных . . . . .	3-11
данных типа REAL . . . . .	3-17
данных типа "двойное слово" (32 бита) . . . . .	3-14
словных данных (16 бит) . . . . .	3-12
Обработка битовых данных в качестве словного операнда . . . . .	3-11
Обработка результата . . . . .	5-15
Обработка строковых величин . . . . .	7-253
Образование	
матрицы ввода . . . . .	6-207
Область программы (активация и деактивация) . . . . .	5-56
Область фиксации . . . . .	A-105
Обмен	
блоками двоичных данных (16 бит) . . . . .	6-144
Обмен данными	
16 бит . . . . .	6-141
32 бита . . . . .	6-141
Обновление	
серий случайных чисел . . . . .	7-387
сетевых и интерфейсных данных . . . . .	6-175
Обновление канала связи . . . . .	6-175
активация . . . . .	6-178
Обновления канала связи	
деактивация . . . . .	6-178

Обновление сетевых данных . . . . .	8-8	Перестановка	
Обозначения, принятые в названиях команд . . . . .	3-3	байтов в пределах двоичного числа . . . . .	6-147
Объединение данных (BTOW) . . . . .	7-110	Пересчет	
Объединение данных (NUNI) . . . . .	7-104	из градусов в радианы . . . . .	7-372
Ограничение выходного значения . . . . .	7-411	из радиан в градусы . . . . .	7-375
Ограничение диапазона выходных значений. . . . .	7-413	Переход к концу программы . . . . .	6-156
Операнды		Поиск . . . . .	7-114
MELSEC A . . . . .	4-3	Поиск	
MELSEC Q . . . . .	4-4	данных . . . . .	7-80
Операнды комментариев. . . . .	7-209	максимальных значений в 16-битных	
Операции записи и считывания		данных . . . . .	7-114
в MELSECNET. . . . .	8-4	максимальных значений в 32-битных	
в MELSECNET(I/II/B) . . . . .	8-4	данных . . . . .	7-114
в MELSECNET/10 . . . . .	8-3	минимальных значений в 16-битных	
Операция		данных . . . . .	7-117
управляемая по фронту . . . . .	5-9	минимальных значений в 32-битных	
Операторы сравнения . . . . .	6-2	данных . . . . .	7-117
D=, D, D>, D=. . . . .	6-9	строковых величин. . . . .	7-341
E=, E, E>, E=. . . . .	6-12	16-битных данных. . . . .	7-80
=, , >, = . . . . .	6-6	32-битных данных. . . . .	7-82
\$=, \$, \$>, \$=. . . . .	6-16	Поиск и устранение ошибок в случае	
Описатель экземпляра. . . . .	A-38	процессоров "System Q" . . . . .	9-7
Определение битовых блоков для словных		Последовательное соединение контактов . . . . .	5-5
данных. . . . .	3-12	Последовательное соединение параллельных схем. . . . .	5-12
Опрос состояния		Преобразование . . . . .	6-103
16 бит. . . . .	7-70	Преобразование	
32 бита. . . . .	7-70	результатов операций . . . . .	5-20
Организующие команды . . . . .	5-55	форматов времени . . . . .	7-456
Особые функции . . . . .	7-351	Преобразование в отрицательное число	
Отладка . . . . .	7-223	16-битных двоичных данных . . . . .	6-109
Ошибки обработки . . . . .	3-31	32-битных двоичных данных	
		(только серия Q) . . . . .	6-109
		Преобразование чисел с плавающей запятой	
		в двоично-десятичный формат . . . . .	7-345
		Преобразование числа с плавающей запятой	
		в десятичный формат . . . . .	7-349
		Прерывание	
		вызов программы прерывания . . . . .	2-27
		Прерывание обработки . . . . .	5-68
		Прерывание цикла FOR-NEXT. . . . .	7-132
		Прерывания	
		активация . . . . .	6-159
		дезактивация . . . . .	6-159
		Прием	
		данных от других станций . . . . .	8-50
		Прикладные команды . . . . .	6-1
		Прикладные команды, часть II . . . . .	7-1
		Применение	
		строковых данных (STRING) . . . . .	3-22
		Принцип действия команды . . . . .	4-6
		Проверка данных операндов. . . . .	3-33
		Проверка на ошибки (CHKST, CHK) . . . . .	7-225
		Проверка области операндов . . . . .	3-31
		Проверочный контур . . . . .	7-229
		Программа. . . . .	12-3
		станции . . . . .	11-157
		Программа микрокомпьютера	
		области памяти центральных процессоров . . . . .	12-1
		Программирование	
		переменных. . . . .	3-7
		специализированные команды . . . . .	3-6
		Программируемый таймер . . . . .	6-188
		Программный компонент (POU). . . . .	3-19

## П

Параллельное соединение контактов . . . . .	5-5	Параллельное соединение последовательных	
Параллельное соединение последовательных		схем. . . . .	5-12
Параллельное соединение с управлением		по фронту. . . . .	5-9
Параметры контроллеров . . . . .	1-3	Параметры маршрутизации (SEND) . . . . .	8-43
Передача . . . . .	11-21	Передача	
блока двоичных данных (16 бит) . . . . .	6-138	блоков двоичных данных (16 бит) . . . . .	6-135
строковых величин . . . . .	6-127	чисел с плавающей запятой . . . . .	6-124
Передача блоков двоичных данных		(процессоры "System Q") . . . . .	9-42
Передача данных (16 бит) . . . . .	6-121	Передача данных (32 бита) . . . . .	6-121
Передача данных в файлы и из файлов в случае		процессоров "System Q". . . . .	9-9
Передача данных между процессорами		Q4ARCPU . . . . .	10-6
Передача данных на другие станции. . . . .	8-43	Переключение	
Переключение		между блоками регистров файлов . . . . .	7-426
между областями MAIN и SUB программы . . . . .	7-151	между файлами комментариев . . . . .	7-432
между файлами регистров файлов . . . . .	7-429	Переменная в команде . . . . .	3-1
Переменная в команде . . . . .	3-1	Переменные в команде . . . . .	4-5
Перемещение		фрагментов строковой величины . . . . .	7-336

Программы микрокомпьютера  
 объем и области памяти. . . . . 12-1  
 применение . . . . . 12-2  
 присвоение адресов . . . . . 12-3  
 разбивка области памяти . . . . . 12-4  
 распределение памяти . . . . . 12-3  
 серия "A" . . . . . 12-1  
 указания по созданию . . . . . 12-2  
 функция . . . . . 12-1  
 Процессоры AnA/AnU . . . . . 1-2  
 Процессоры AnAS/AnUS . . . . . 1-2  
 Процессоры AnN. . . . . 1-2  
 Процессоры AnS . . . . . 1-2  
 Процессоры QnA. . . . . 1-2  
 Процессоры "System Q" . . . . . 1-2  
 Прочие команды . . . . . 5-67, 6-180  
 Прочие прикладные команды . . . . . 7-477  
 Пустые шаги в программе . . . . . 5-71

**Р**

Разветвления программы . . . . . 6-149  
 Разделение  
 и объединение групп байтов . . . . . 7-108  
 16-битных данных . . . . . 7-97  
 Разделение данных . . . . . 7-103  
 Распознавание и устранение ошибок (обзор) . . . . . 2-46  
 Расчет шагов программы . . . . . 3-39  
 Регистр файлов  
 непосредственная запись . . . . . 7-490  
 непосредственное считывание. . . . . 7-486  
 Регистры, относящиеся к командам . . . . . A-102  
 Регистры файлов (количество шагов  
 программы при обработке) . . . . . 3-37  
 Режим  
 низкой скорости обработки . . . . . 7-476  
 одного выполнения программы на цикл . . . . . 7-474  
 Режим с квитированием при командах  
 коммуникации . . . . . 8-30  
 Резервный режим  
 команда переключения . . . . . 7-470  
 со сбросом выходов . . . . . 7-472  
 Релейная станция . . . . . 8-43

**С**

Сброс  
 выходов в подпрограммах . . . . . 7-140  
 выходов в подпрограммах (EFCALL) . . . . . 7-147  
 маркеров ошибки и светодиодной  
 индикации . . . . . 7-220  
 областей битов . . . . . 7-74  
 операнда . . . . . 5-37  
 сторожевых таймеров . . . . . 7-479  
 Сдвиг  
 битовых операндов. . . . . 5-53  
 влево (BSFL) . . . . . 7-60  
 влево (DSFL) . . . . . 7-63  
 влево (SFL) . . . . . 7-57  
 вправо (DSFR) . . . . . 7-63  
 вправо (SFR) . . . . . 7-57  
 n битовых операндов на 1 бит . . . . . 7-60  
 n словных операндов на 1 адрес. . . . . 7-63  
 16-битного слова данных на n битов . . . . . 7-57

Системная информация . . . . . A-47  
 Системные такты и счетчики . . . . . A-52  
 Системные часы и счетчики. . . . . A-96  
 Словные операнды . . . . . 3-13  
 Словные операнды в случае данных типа  
 "двойное слово" . . . . . 3-16  
 Сложение данных времени . . . . . 7-446  
 Сложение и вычитание  
 двоично-десятичных данных (4-значных) . . . . . 6-44  
 двоично-десятичных данных (8-значных) . . . . . 6-49  
 двоичных данных . . . . . 6-69  
 двоичных данных (16 бит) . . . . . 6-29  
 двоичных данных (32 бита) . . . . . 6-34  
 чисел с плавающей запятой. . . . . 6-61  
 Случайные числа  
 генерирование. . . . . 7-387  
 Смещение входных 16-битных и 32-битных  
 двоичных данных . . . . . 7-417  
 Смещение выходных 16-битных  
 и 32-битных двоичных данных . . . . . 7-421  
 Содержимое индексных регистров  
 восстановление . . . . . 7-501  
 фиксация . . . . . 7-501  
 Соединение  
 строковых величин . . . . . 6-73  
 Соединяющие команды. . . . . 5-11  
 Создание проверочных контуров . . . . . 7-241  
 Сортировка  
 16-битных данных. . . . . 7-120  
 32-битных данных. . . . . 7-121  
 Сохранение определенных фрагментов  
 строковой величины. . . . . 7-335  
 Специализированные команды коммуникации . . . . . 8-4  
 Специализированные команды коммуникации  
 (серия QnA) . . . . . 8-11  
 Специализированные команды  
 программирования . . . . . 3-6  
 Специальные маркеры  
 для областей фиксации . . . . . A-60  
 Специальные маркеры (серии "A") . . . . . A-68  
 Специальные маркеры для устранения ошибок . . . . . A-58  
 Специальные регистры  
 для указания модулей  
 с неисправным предохранителем . . . . . A-106  
 для устранения ошибок. . . . . A-104  
 карты памяти . . . . . A-99  
 Сравнение  
 команд индикации . . . . . A-40  
 строковых величин . . . . . 6-16  
 центральных процессоров . . . . . A-33  
 чисел с плавающей запятой. . . . . 6-12  
 Сравнение 16-битных данных . . . . . 6-6  
 Сравнение 32-битных данных . . . . . 6-9  
 Стандартная библиотека (Standard\_Lib) . . . . . 3-3  
 Стирание  
 определенных блоков данных . . . . . 7-184  
 программы из памяти для хранения  
 программ. . . . . 9-37  
 Стирание и загрузка программы . . . . . 9-39  
 Стирание ошибки (модуль Ethernet) . . . . . 11-64  
 Строблирующий сигнал . . . . . 7-209  
 Структура . . . . . 12-3  
 Сумматор. . . . . 7-220

Суммирование	
логическое . . . . .	7-21
16-битных двоичных данных . . . . .	7-124
32-битных двоичных данных . . . . .	7-126
Счетчик импульсов . . . . .	6-201
Считывание . . . . .	11-8
Считывание	
введенных данных . . . . .	7-176
данных времени . . . . .	7-436
данных других станций (READ) . . . . .	8-16
данных других станций (SREAD) . . . . .	8-23
данных других станций (ZNRD) . . . . .	8-79
данных из локальной станции . . . . .	8-87
данных из специальных модулей . . . . .	7-191
данных из удаленной станции . . . . .	8-96
данных комментариев . . . . .	7-286
данных, введенных последними . . . . .	7-180
информации маршрутизации . . . . .	8-108
1-словных данных (16 бит) из специальных модулей . . . . .	7-191
2-словных данных (32 бита) из специальных модулей . . . . .	7-191
Считывание данных	
в программе прерывания . . . . .	11-41
из другой станции (CC-Link) . . . . .	11-112
из интеллектуальной станции CC-Link . . . . .	11-145
из интеллектуальной станции (CC-Link) . . . . .	11-139
из модуля Profibus . . . . .	11-29
из модуля интерфейса . . . . .	11-4
из фиксированных буферов . . . . .	11-37
с помощью команды RIRD . . . . .	11-119
Считывание данных из общей области памяти . . . . .	9-50
Считывание информации модуля . . . . .	9-2
Считывание кода ошибки из модуля Ethernet . . . . .	11-70
Считывание/запись информации маршрутизации . . . . .	8-106

**Т**

Таблица процессоров в описании команд . . . . .	4-2
Таймер	
высокоскоростной . . . . .	6-190
для специальных функций . . . . .	6-190
низкоскоростной . . . . .	6-190
Тип данных	
DINT . . . . .	3-19
DWORD . . . . .	3-19
Типы данных . . . . .	3-9
Типы центральных процессоров . . . . .	1-2
Трассировка программы . . . . .	7-223
выполнение . . . . .	7-252
сброс . . . . .	7-252
установка . . . . .	7-252

**У**

Удаленные станции ввода-вывода . . . . .	8-67
Указание конца программы . . . . .	5-61
Умножение	
логическое . . . . .	7-12
Умножение и деление	
двоично-десятичных данных (4-значных) . . . . .	6-54
двоично-десятичных данных (8-значных) . . . . .	6-57
двоичных данных (16 бит) . . . . .	6-37
двоичных данных (32 бита) . . . . .	6-41
чисел с плавающей запятой . . . . .	6-66

Условия выполнения	
для обновления канала связи . . . . .	6-178
Условный переход	
при следующем прогоне . . . . .	6-151
Установка . . . . .	11-14
Установка	
контакта счетчика . . . . .	5-29
маркера фронта . . . . .	5-21
операнда . . . . .	5-35
Установка и сброс	
маркеров ошибки (серии "Q" и "System Q") . . . . .	5-40
маркеров ошибки (серия "A") . . . . .	5-41
отдельных битов . . . . .	7-67
Установление связи (Ethernet) . . . . .	11-51
Устранение ошибок	
специальные маркеры . . . . .	A-58
специальные регистры . . . . .	A-104

**Ф**

Фиксация состояния	
сброс . . . . .	7-246
установка . . . . .	7-246
Фиктивные контакты . . . . .	7-242
команд . . . . .	3-34
отображение . . . . .	2-5
программ прерывания (только серии "Q" и "System Q") . . . . .	6-160
Флаг переноса	
сброс . . . . .	7-481
установка . . . . .	7-481
Фрагмент строковых данных	
слева . . . . .	7-332
справа . . . . .	7-331

**Ц**

Цель перехода . . . . .	6-154
Цикл FOR - NEXT . . . . .	7-129

**Ш**

Широтно-импульсная модуляция . . . . .	6-205
--	-------

**Э**

Экспоненциальное . . . . .	7-316
Экспоненциальное представление	
числа с плавающей запятой . . . . .	7-315
числа типа REAL . . . . .	7-309

**Символы**

Сдвиг	
вправо (BSFR) . . . . .	7-60
Частичное обновление ввода-вывода	
(серия "A") . . . . .	6-171
(серии "Q" и "System Q") . . . . .	6-169
> . . . . .	6-5
\$MOV, \$MOV.P . . . . .	6-126
\$+, \$+P . . . . .	6-72



**MITSUBISHI ELECTRIC EUROPE B.V. /// РОССИЯ /// Москва /// Космодамианская наб. 52, стр. 3**  
**Тел.: +7 495 721-2070 /// Факс: +7 495 721-2071 /// [automation@mer.mee.com](mailto:automation@mer.mee.com) /// [www.mitsubishi-automation.ru](http://www.mitsubishi-automation.ru)**