

## Замечания.

Изделия корпорации OMRON производятся для эксплуатации квалифицированным персоналом, использующим соответствующие приемы, и только для целей, описанных в настоящем Руководстве.

Для обозначения и классификации мер предосторожностей в настоящем Руководстве используются следующие ниже обозначения. Пренебрежение мерами предосторожности может стать причиной травмирования персонала или повреждения оборудования.

**Опасность!** *Обозначает информацию, пренебрежение которой может стать причиной смертельной травмы или серьезного ранения.*

**Предупреждение!** *Обозначает информацию, пренебрежение которой, вероятно, может стать причиной смертельной травмы или серьезного ранения.*

**Внимание!** *Обозначает информацию, пренебрежение которой может стать причиной тяжелых или незначительных травм персонала, повреждения оборудования или сбоев в его работе.*

## Ссылка на продукцию OMRON

Все изделия фирмы OMRON в настоящем руководстве пишутся с заглавной буквы. Слово «Модуль» начинается с заглавной буквы, если он является продукцией OMRON вне зависимости от того, является ли это именем собственным изделия.

Аббревиатура Ch, которая появляется на некоторых дисплеях изделий фирмы OMRON, часто означает «Слово» и в этом смысле в документации обозначается аббревиатурой «Wd»

Аббревиатура «PC» обозначает программируемый контролер и не используется в качестве аббревиатуры для другого оборудования.

## Визуальная помощь.

Следующие ниже подписи появляются в левой стороне страницы Руководства для определения различных типов информации.

**Примечание:** *Обозначает интересную информацию, необходимую для эффективного и удобного управления изделием*

**1,2,3...** *Обозначает перечни разных типов, например перечень операции, перечень проверок и т.д.*



## О настоящем Руководстве

В настоящем Руководстве приводится описание команд релейно-контактных программ, поддерживаемых Модулями центрального процессора CS1G/H-CPU -EV1, для Программируемых контроллеров серии CS1. Руководство содержит разделы, указанные ниже.

Непрерывно внимательно изучите настоящее Руководство и относящиеся к нему другие Руководства по эксплуатации, перечисленные в следующей ниже таблице. Убедитесь в том, что перед осуществлением попытки запуска и эксплуатации Модулей центрального процессора CS1G/H-CPU \_\_-EV1 в системах программируемых контроллеров Вы поняли всю изложенную информацию.

Наименование	Каталог №	Содержание
Программируемые контроллеры CS1G/H-CPU -Е серии SYSMAC CS1. Руководство по программированию.	W340	Описывает команды релейно-контактных программ, поддерживаемые Программируемыми контроллерами серии CS1. (Настоящее руководство).
Программируемые контроллеры CS1G/H-CPU -Е серии SYSMAC CS1. Руководство по эксплуатации.	W339	Описывает порядок запуска и эксплуатации Программируемых контроллеров серии CS1.
Пульты программирования C200H -PRO27-Е, CQM1-PRO01-Е серии SYSMAC CS1. Руководство по эксплуатации.	W341	Описывает порядок программирования и управления Программируемым контроллером серии CS1 при помощи Пульта программирования.
Коммуникационные команды CS1G/H -CPU -EV1, CS1W-SCB21/41, CS1W-CSU21 серии SYSMAC CS1. Справочное руководство.	W342	Описывает команды коммуникационного обмена серии С (Host Link) и FINS команды, используемые Программируемыми контроллерами серии CS1.
СХ- программатор SYSMAC WS02-CXP -Е. Руководство по эксплуатации.	-	Содержит информацию об использовании СХ-программатора, устройства программирования, поддерживающего работу Программируемых контроллеров серии CS1.
Платы последовательного коммуникационного обмена и Модули последовательного коммуникационного обмена CS1W-SCB21/41, CS1W-SCU21 серии SYSMAC CS1. Руководство по эксплуатации.	W336	Описывает порядок совместной эксплуатации Платы последовательного коммуникационного обмена и Модулей последовательного коммуникационного обмена с внешними устройствами, а также использование стандартных системных протоколов для продукции корпорации OMRON.
СХ- протокол SYSMAC WS02-PSTC1-Е. Руководство по применению.	W344	Описывает порядок использования СХ-протокола для создания макро- протокола, применяемого для осуществления коммуникационного обмена с внешними устройствами.
Модуль Ethernet CS1W-ETN01 серии SYSMAC CS1. Руководство по эксплуатации.	W343	Описывает порядок запуска и эксплуатации Модуля Ethernet CS1W-ETN01.

### Раздел 1

Представляет Программируемые контроллеры серии CS1 в отношении набора команд, поддерживаемых контроллерами.

### Раздел 2

Содержит перечни команд, которые при эксплуатации могут использоваться для справки.

### Раздел 3

Содержит описание отдельных команд, используемых в контроллерах серии CS1.

### Раздел 4

Содержит данные о времени выполнения команд, а также количество шагов для каждой из команд контроллеров серии CS1.

**Внимание!** Невнимательное изучение и непонимание информации, изложенной в настоящем Руководстве, может привести к травмированию обслуживающего персонала, в том числе и со смертельным исходом, повреждению оборудования или его разрушению. Тщательно изучите каждый раздел настоящего Руководства. Перед осуществлением попыток выполнения любых операций убедитесь в том, что Вы полностью понимаете изложенную в данном разделе и относящуюся к этому разделу информацию.



## Contents

<b>Меры предосторожности.</b>	<b>15</b>
1. Категории персонала	16
2. Общие меры предосторожности	16
3. Меры предосторожности для обеспечения безопасности	16
4. Меры предосторожности при размещении	17
5. Меры предосторожности при эксплуатации	17
6. Соответствие Директивам ЕС	21
6-1 Применяемые Директивы	21
6-2 Общие представления	21
6-3 Соответствие Директивам ЕС	21
6-4 Методы снижения помех на релейном выходе	22
<b>Глава 1</b>	
<b>Введение.</b>	<b>25</b>
1-1 Основные характеристики команд	26
1-1-1 Объем программы	26
1-1-2 Дифференцированные команды	26
1-1-3 Модификации команд	27
1-1-4 Расположение команды и условия выполнения	27
1-1-5 Ввод данных в операнды	28
1-1-6 Форматы данных	32
1-2 Проверки выполнения команд	33
1-2-1 Ошибки, возникающие в процессе выполнения команд	33
1-2-2 Критические ошибки (ошибки программы)	34
<b>Глава 2</b>	
<b>Краткое описание команд</b>	<b>37</b>
2-1 Функциональная классификация команд	38
2-2 Функции команд	43
2-2-1 Команды последовательного ввода	43
2-2-2 Команды последовательного вывода	44
2-2-3 Команды последовательного управления	45
2-2-4 Команды управления таймером и счетчиком	46
2-2-5 Команды сравнения	47
2-2-6 Команды перемещения данных	48
2-2-7 Команды сдвига данных	49
2-2-8 Команды увеличения, уменьшения	50
2-2-9 Символьные математические команды	50
2-2-10 Команды преобразования	51
2-2-11 Логические команды	52
2-2-12 Специальные математические команды	53
2-2-13 Команды математических операций с плавающей запятой	53
2-2-14 Команды обработки табличных данных	55
2-2-15 Команды управления данными	55
2-2-16 Команды подпрограмм	56
2-2-17 Команды управления прерываниями	56
2-2-18 Шаговые команды	56
2-2-19 Команды Базовых модулей ввода/вывода	56
2-2-20 Команды последовательного коммуникационного обмена	57
2-2-21 Сетевые команды	57
2-2-22 Команды памяти файлов	57
2-2-23 Команды дисплея	57
2-2-24 Команды управления часами	58
2-2-25 Команды отладки программы	58

2-2-26	Команды диагностики отказов . . . . .	58
2-2-27	Прочие команды . . . . .	58
2-2-28	Команды программирования блоков . . . . .	58
2-2-29	Команды обработки текстовых строк программы . . . . .	60
2-2-30	Команды управления задачами . . . . .	61
2-3	Алфавитный список команд согласно их мнемоническому изображению . . . . .	61
2-4	Перечень команд по функциональному коду . . . . .	72
<b>Глава 3 Команды . . . . .</b>		<b>81</b>
3-1	Система обозначений и расположение описания команды . . . . .	82
3-2	Дополнительные возможности для версии 1 Модуля центрального процессора серии CS1 . . . . .	84
3-3	Команды последовательного ввода . . . . .	85
3-3-1	LOAD: LD . . . . .	85
3-3-2	LOAD NOT: LD NOT . . . . .	87
3-3-3	AND: AND . . . . .	89
3-3-4	AND NOT: AND NOT . . . . .	90
3-3-5	OR: OR . . . . .	92
3-3-6	OR NOT: OR NOT . . . . .	93
3-3-7	AND LOAD: AND LD . . . . .	95
3-3-8	OR LOAD: OR LD . . . . .	97
3-3-9	Дифференцированные команды и команды немедленной регенерации . . . . .	99
3-3-10	Временная диаграмма выполнения команд ввода/ вывода . . . . .	99
3-3-11	Биты области TR . . . . .	100
3-3-12	NOT: NOT(520) . . . . .	102
3-3-13	Условие ON/OFF : UP(521) и DOWN(522). . . . .	103
3-3-14	Команда BIT TEST: TST(350) и TSTN(351) . . . . .	104
3-4	Команды последовательного вывода . . . . .	107
3-4-1	OUTPUT: OUT . . . . .	107
3-4-2	OUTPUT NOT: OUT NOT . . . . .	108
3-4-3	KEEP: KEEP(011) . . . . .	109
3-4-4	Команда дифференцирования вверх/вниз DIFFERENTIATE UP/DOWN: DIFU(013) и DIFD(014) . . . . .	113
3-4-5	Команды SET и RESET: SET и RSET . . . . .	115
3-4-6	Команда установки и сброса множества битов MULTIPLE BIT SET/RESET: SETA(530)/RSTA(531) . . . . .	117
3-5	Команды последовательного управления . . . . .	120
3-5-1	Команда завершения программы END: END(001) . . . . .	120
3-5-2	Команда NO OPERATION: NOP(000). . . . .	121
3-5-3	Команды блокирования Interlock and Interlock Clear . . . . .	121
3-5-4	Команды выполнения перехода JUMP и JUMP END: JMP(004) и JME(005) . . . . .	125
3-5-5	Команды условного перехода CONDITIONAL JUMP: CJP(510)/CJPN(511) . . . . .	128
3-5-6	Команды выполнения переходов MULTIPLE JUMP и JUMP END: JMP0(515) и JME(516) . . . . .	131
3-5-7	Циклы FOR-NEXT: FOR(512)/NEXT(513) . . . . .	133
3-5-8	Команда отмены цикла BREAK LOOP: BREAK(514) . . . . .	137
3-6	Команды управления таймерами и счетчиками . . . . .	138
3-6-1	Команда запуска таймера TIMER: TIM . . . . .	139
3-6-2	Команда запуска высокоскоростного таймера HIGH-SPEED TIMER: TIMH(015). . . . .	142
3-6-3	Команда запуска миллисекундного таймера ONE-MS TIMER: TMH . . . . .	145
3-6-4	Команда запуска аккумулятивного таймера ACCUMULATIVE TIMER: TTIM(087) . . . . .	147
3-6-5	Команда запуска долговременного таймера LONG TIMER: TIML(542) . . . . .	151
3-6-6	Команда запуска таймера с несколькими выводами MULTI-OUTPUT TIMER MTIM(543). . . . .	153
3-6-7	Команда запуска счетчика COUNTER: CNT . . . . .	158
3-6-8	Команда запуска реверсивного счетчика REVERSIBLE COUNTER: CNTR(012) . . . . .	160
3-6-9	Команда выполнения переустановки таймера/счетчика RESET TIMER/COUNTER: CNR(545). . . . .	164
3-6-10	Пример использования таймера и счетчика . . . . .	166
3-6-11	Косвенное указание номеров таймеров/счетчиков . . . . .	171

3-7	Команды сравнения . . . . .	175
3-7-1	Команды сравнения ввода (300...328) . . . . .	175
3-7-2	Команда сравнения COMPARE: CMP(020). . . . .	180
3-7-3	Команда сравнения данных двойной длины DOUBLE COMPARE: CMPL(060). . . . .	182
3-7-4	Команда сравнения двоичных данных со знаком SIGNED BINARY COMPARE: CPS(114). . . . .	185
3-7-5	Команда сравнения данных двойной длины DOUBLE COMPARE: CPSL(115) . . . . .	187
3-7-6	Команда сравнения последовательностей слов MULTIPLE COMPARE: MCOMP(019) . . . . .	190
3-7-7	Команда сравнения табличных данных TABLE COMPARE: TCMP(085) . . . . .	192
3-7-8	Команда сравнения блока BLOCK COMPARE: BCMP(068) . . . . .	195
3-8	Команды перемещения данных . . . . .	198
3-8-1	Команда MOVE: MOV(021) . . . . .	198
3-8-2	Команда MOVE NOT: MVN(022) . . . . .	199
3-8-3	Команда перемещения двойного слова DOUBLE MOVE: MOVL(489). . . . .	200
3-8-4	Команда передачи двойного дополнения DOUBLE MOVE NOT: MVNL(499). . . . .	202
3-8-5	Команда перемещения бита MOVE BIT: MOV(082) . . . . .	203
3-8-6	Команда перемещения числа MOVE DIGIT: MOVD(083). . . . .	205
3-8-7	Команда передачи множества битов MULTIPLE BIT TRANSFER: XFRB(062) . . . . .	208
3-8-8	Команда передачи данных блока BLOCK TRANSFER: XFER(070). . . . .	210
3-8-9	Команда задания блока BLOCK SET: BSET(071) . . . . .	212
3-8-10	Команда обмена данными DATA EXCHANGE: XCHG(073). . . . .	214
3-8-11	Команда обмена двойными данными DOUBLE DATA EXCHANGE: XCGL(562) . . . . .	215
3-8-12	Команда распределения одного слова SINGLE WORD DISTRIBUTE: DIST(080) . . . . .	217
3-8-13	Команда отбора данных DATA COLLECT: COLL(081). . . . .	219
3-8-14	Команда перемещения в регистр MOVE TO REGISTER: MOV(560) . . . . .	221
3-8-15	Команда записи MOVE TIMER/COUNTER PV TO REGISTER: MOV(561) . . . . .	223
3-9	Команды смещения данных . . . . .	225
3-9-1	Команда сдвига регистра SHIFT REGISTER: SFT(010) . . . . .	225
3-9-2	Команда реверсивного смещения регистра REVERSIBLE SHIFT REGISTER: SFTR(084) . . . . .	227
3-9-3	Команда ASYNCHRONOUS SHIFT REGISTER: ASFT(017) . . . . .	229
3-9-4	Команда смещения слова WORD SHIFT : WSFT(016) . . . . .	232
3-9-5	Команда арифметического смещения влево ARITHMETIC SHIFT LEFT: ASL(025) . . . . .	233
3-9-6	Команда смещения двойных данных DOUBLE SHIFT LEFT: ASLL(570) . . . . .	235
3-9-7	Команда арифметического смещения вправо ARITHMETIC SHIFT RIGHT: ASR(026). . . . .	237
3-9-8	Команда смещения двойных данных вправо DOUBLE SHIFT RIGHT: ASRL(571) . . . . .	238
3-9-9	Команда вращения влево ROTATE LEFT: ROL(027) . . . . .	240
3-9-10	Команда вращения двойных данных влево DOUBLE ROTATE LEFT: ROLL(572) . . . . .	242
3-9-11	Команда вращения вправо ROTATE RIGHT: ROR(028) . . . . .	243
3-9-12	Команда вращения двойных данных вправо DOUBLE ROTATE RIGHT: RORL(573) . . . . .	245
3-9-13	Команда вращения влево без переноса ROTATE LEFT WITHOUT CARRY: RLNC(574) . . . . .	247
3-9-14	Команда вращения двойных данных влево без переноса DOUBLE ROTATE LEFT WITHOUT CARRY: RLNL(576) . . . . .	248
3-9-15	Команда вращения вправо без переноса ROTATE RIGHT WITHOUT CARRY: RRNC(575). . . . .	250
3-9-16	Команда вращения двойных данных вправо без переноса DOUBLE ROTATE RIGHT WITHOUT CARRY: RRNL(577) . . . . .	251
3-9-17	Команда перемещения на одну цифру влево ONE DIGIT SHIFT LEFT: SLD(074) . . . . .	253
3-9-18	Команда перемещения на одну цифру вправо ONE DIGIT SHIFT RIGHT: SRD(075) . . . . .	255
3-9-19	Команда смещения N битов данных влево SHIFT N-BIT DATA LEFT: NSFL(578). . . . .	256
3-9-20	Команда смещения N битов данных вправо SHIFT N-BIT DATA RIGHT: NSFR(579) . . . . .	258
3-9-21	Команда смещения данных на N битов влево SHIFT N-BIT LEFT: NASL(580). . . . .	260
3-9-22	Команда смещения двойных данных на N битов влево DOUBLE SHIFT N-BIT LEFT: NSLL(582) . . . . .	263
3-9-21	Команда смещения данных на N битов вправо SHIFT N-BIT RIGHT: NASR(581) . . . . .	266
3-9-24	Команда смещения двойных данных на N битов вправо DOUBLE SHIFT N-BIT RIGHT: NSRL(583) . . . . .	269
3-10	Команды увеличения/уменьшения . . . . .	272
3-10-1	Команда увеличения двоичного числа INCREMENT BINARY: ++(590) . . . . .	272
3-10-2	Команда увеличения двойного двоичного числа DOUBLE INCREMENT BINARY: ++L(591) . . . . .	273



3-10-3	Команда уменьшения двоичного числа DECREMENT BINARY:-- (592)	275
3-10-4	Команда уменьшения двойного двоичного числа DOUBLE DECREMENT BINARY:--L(593)	277
3-10-5	Команда увеличения двоично-десятичного числа INCREMENT BCD:++B(594)	279
3-10-6	Команда увеличения двойного двоично-десятичного числа DOUBLE INCREMENT BCD:++BL(595)	281
3-10-7	Команда уменьшения двоично-десятичного числа DECREMENT BCD:--B(596)	283
3-10-8	Команда уменьшения двойного двоично-десятичного числа DOUBLE DECREMENT BCD:--BL(597)	285
3-11	Символьные математические команды	288
3-11-1	Команда сложения без переноса двоичных данных со знаком SIGNED BINARY ADD WITHOUT CARRY:+(400)	288
3-11-2	Команда сложения без переноса двойных двоичных данных со знаком DOUBLE SIGNED BINARY ADD WITHOUT CARRY:+L(401)	290
3-11-3	Команда сложения двоичных данных со знаком с переносом SIGNED BINARY ADD WITH CARRY:+C(402)	292
3-11-4	Команда сложения двойных двоичных данных со знаком с переносом DOUBLE SIGNED BINARY ADD WITH CARRY:+CL(403)	294
3-11-5	Команда сложения без переноса двоично-десятичных данных BCD ADD WITHOUT CARRY:+B(404)	295
3-11-6	Команда сложения без переноса двойных двоично-десятичных данных DOUBLE BCD ADD WITHOUT CARRY:+BL(405)	297
3-11-7	Команда сложения двоично-десятичных данных с переносом BCD ADD WITH CARRY:+BC(406)	299
3-11-8	Команда сложения двойных двоично-десятичных данных с переносом DOUBLE BCD ADD WITH CARRY:+BCL(407)	300
3-11-9	Команда вычитания двоичных данных со знаком без переноса SIGNED BINARY SUBSTRACT WITHOUT CARRY:-(410)	302
3-11-10	Команда вычитания двойных двоичных данных со знаком без переноса DOUBLE SIGNED BINARY SUBSTRACT WITHOUT CARRY:-L(411)	304
3-11-11	Команда вычитания двоичных данных со знаком с переносом SIGNED BINARY SUBSTRACT WITH CARRY:-C(412)	307
3-11-12	Команда вычитания с переносом двойных двоичных данных со знаком DOUBLE SIGNED BINARY SUBSTRACT WITH CARRY:-CL(413)	309
3-11-13	Команда вычитания двоично-десятичных данных без переноса BCD SUBSTRACT WITHOUT CARRY:-B(414)	312
3-11-14	Команда вычитания двойных двоично-десятичных данных без переноса DOUBLE BCD SUBSTRACT WITHOUT CARRY:-BL(415)	313
3-11-15	Команда вычитания двоично-десятичных данных с переносом BCD SUBSTRACT WITH CARRY:-BC(416)	316
3-11-16	Команда вычитания с переносом двойных двоично-десятичных данных DOUBLE BCD SUBSTRACT WITH CARRY:-BCL(417)	318
3-11-17	Команда перемножения двоичных данных со знаком SIGNED BINARY MULTIPLY:*(420)	320
3-11-18	Команда перемножения двойных двоичных данных со знаком DOUBLE SIGNED BINARY MULTIPLY:*L(421)	321
3-11-19	Команда перемножения двоичных данных без знака UNSIGNED BINARY MULTIPLY:*U(422)	323
3-11-20	Команда перемножения двойных двоичных данных без знака DOUBLE UNSIGNED BINARY MULTIPLY:*UL(423)	324
3-11-21	Команда перемножения двоичных данных без знака UNSIGNED BINARY MULTIPLY:*U(422)	326
3-11-22	Команда перемножения двойных двоично-десятичных данных DOUBLE BCD MULTIPLY:*BL(425)	327
3-11-23	Команда деления двоичных данных со знаком SIGNED BINARY DIVIDE:/(430)	329
3-11-24	Команда деления двойных двоичных данных со знаком DOUBLE SIGNED BINARY DIVIDE:/L(431)	330
3-11-25	Команда деления двоичных данных без знака UNSIGNED BINARY DIVIDE:/U(432)	332
3-11-26	Команда деления двойных двоичных данных без знака DOUBLE UNSIGNED BINARY DIVIDE:/U(433)	334



3-11-27	Команда деления двоично-десятичных BCD DIVIDE: /B(434)	335
3-11-28	Команда деления двойных двоично-десятичных данных DOUBLE BCD DIVIDE: /BL(435)	337
3-12	Команды преобразования	339
3-12-1	Команда преобразования двоично-десятичных данных в двоичные данные BCD-TO- BINARY: BIN(023)	339
3-12-2	Команда преобразования двойных двоично-десятичных данных в двойные двоичные данные DOUBLE BCD-TO- DOUBLE BINARY: BINL(058)	340
3-12-3	Команда преобразования двоичных данных в двоично-десятичные данные BINARY-TO-BCD: BCD(024)	342
3-12-4	Команда преобразования двойных двоичных данных в двойные двоично-десятичные данные DOUBLE BINARY -TO- DOUBLE BCD: BCDL(059)	343
3-12-5	Команда вычисления дополнения по модулю 2: 2'S COMPLEMENT: NEG(160)	345
3-12-6	Команда вычисления дополнения к двойному слову данных по модулю 2: DOUBLE 2'S COMPLEMENT: NEGL(161)	347
3-12-7	Команда преобразования 16-битового числа в его 32-х битовый эквивалент 16-BIT TO 32 - BIT SIGNED BINARY: SIGN(600)	349
3-12-8	Команда декодирования данных DATA DECODER: MLPX(076)	350
3-12-9	Команда кодирования данных DATA ENCODER: DMPX(077)	355
3-12-10	Команда преобразования данных в ASCII код ASCII CONVERT: ASC(086)	359
3-12-11	Команда преобразования ASCII данных в шестнадцатеричные данные ASCII TO HEX: HEX(162)	363
3-12-12	Команда преобразования данных из колонки в строку COLUMN TO LINE: LINE (063)	367
3-12-13	Команда преобразования данных из строки в колонку LINE TO COLUMN: COLM (064)	369
3-12-14	Команда преобразования двоично-десятичных данных со знаком в двоичные данные со знаком SIGNED BCD-TO-BINARY: BINS(470)	371
3-12-15	Команда преобразования двойных двоично-десятичных данных со знаком в двойные двоичные данные со знаком DOUBLE SIGNED BCD-TO-BINARY: BISL(472)	374
3-12-16	Команда преобразования двоичных данных со знаком в двоично-десятичные данные со знаком SIGNED BINARY-TO-BCD: BCDS(471)	377
3-12-17	Команда преобразования двойных двоичных данных со знаком в двойные двоично-десятичные данные со знаком DOUBLE SIGNED BINARY-TO-BCD: BDSL(473)	379
3-13	Логические команды	384
3-13-1	Команда выполнения логической операции «И» LOGICAL AND: ANDW(034)	384
3-13-2	Команда выполнения логической операции «И» с двойными данными DOUBLE LOGICAL AND: ANDL(610)	385
3-13-3	Команда выполнения логической операции «ИЛИ» LOGICAL OR: ORW(035)	387
3-13-4	Команда выполнения логической операции «ИЛИ» с двойными данными DOUBLE LOGICAL OR: ORWL(611)	388
3-13-5	Команда выполнения логической операции «Исключающее ИЛИ» EXCLUSIVE OR: XORW(036)	390
3-13-6	Команда выполнения логической операции «Исключающее ИЛИ» с двойными данными DOUBLE EXCLUSIVE OR: XORL(612)	391
3-13-7	Команда выполнения логической операции «Исключающее ИЛИ-НЕ» EXCLUSIVE NOR: XNRW(037)	393
3-13-8	Команда выполнения логической операции «Исключающее ИЛИ-НЕ» с двойными данными DOUBLE EXCLUSIVE NOR: XNRL(613)	395
3-13-9	Команда вычисления дополнения COMPLEMENT: COM(029)	397
3-13-10	Команда вычисления дополнения к двойному слову DOUBLE COMPLEMENT: COML(614)	398
3-14	Специальные математические команды	400
3-14-1	Команда вычисления квадратного корня из двоичного числа BINARY ROOT: ROTB(620)	400
3-14-2	Команда вычисления квадратного корня из двоично-десятичного числа BCD SQUARE ROOT: ROOT (072)	401
3-14-3	Команда выполнения математических операций ARITHMETIC PROCESS: APR(069)	405
3-14-4	Команда выполнения деления чисел с плавающей запятой FLOATING POINT DIVIDE: FDIV(079)	410
3-14-5	Команда подсчета битов BIT COUNTER: BCNT(067)	414

3-15	Команды математических операций с плавающей запятой . . . . .	416
3-15-1	Команда преобразования данных с плавающей запятой в двоичные данные FLOATING TO 16-BIT: FIX(450) . . . . .	421
3-15-2	Команда преобразования данных с плавающей запятой в двойные двоичные данные FLOATING TO 32-BIT: FIXL(451) . . . . .	422
3-15-3	Команда преобразования двоичных данных в данные с плавающей запятой 16-BIT TO FLOATING: FLT(452) . . . . .	424
3-15-4	Команда преобразования двоичных данных в данные с плавающей запятой 32-BIT TO FLOATING: FLTL(453) . . . . .	425
3-15-5	Команда выполнения сложения двух чисел с плавающей запятой FLOATING-POINT ADD: +F(454) . . . . .	427
3-15-6	Команда выполнения вычитания двух чисел с плавающей запятой FLOATING-POINT SUBTRACT: -F(455) . . . . .	428
3-15-7	Команда выполнения умножения двух чисел с плавающей запятой FLOATING-POINT MULTIPLY: *F(456) . . . . .	430
3-15-8	Команда выполнения деления двух чисел с плавающей запятой FLOATING-POINT DIVIDE: /F(457) . . . . .	432
3-15-9	Команда преобразования градусов в радианы DEGREES TO RADIANS: RAD(458) . . . . .	434
3-15-10	Команда преобразования радианов в градусы RADIANS TO DEGREES: DEG(459) . . . . .	436
3-15-11	Команда вычисления синуса SINE: SIN(460) . . . . .	437
3-15-12	Команда вычисления косинуса COSINE: COS(461) . . . . .	439
3-15-13	Команда вычисления тангенса TANGENT: TAN(462) . . . . .	440
3-15-14	Команда вычисления арксинуса ARC SINE: ASIN(463) . . . . .	442
3-15-15	Команда вычисления арккосинуса ARC COSINE: ACOS(464) . . . . .	444
3-15-16	Команда вычисления арктангенса ARC TANGENT: ATAN(465) . . . . .	446
3-15-17	Команда вычисления квадратного корня SQUARE ROOT: SQRT(466) . . . . .	447
3-15-18	Команда вычисления натуральной экспоненты EXPONENT: EXP(467) . . . . .	449
3-15-19	Команда вычисления логарифма LOGARITHM: LOG(468) . . . . .	451
3-15-20	Команда возведения в степень EXPONENTIAL POWER: PWR(840) . . . . .	453
3-16	Команды для обработки табличных данных . . . . .	455
3-16-1	Команда задания стека SET STACK: SSET(630) . . . . .	458
3-16-2	Команда записи данных PUSH ONTO STACK: PUSH(632) . . . . .	460
3-16-3	Команда чтения первой записи FIRST IN FIRST OUT: FIFO(633) . . . . .	463
3-16-4	Команда чтения последней записи LAST IN FIRST OUT: LIFO(634) . . . . .	465
3-16-5	Команда задания таблицы записей DIMENSION RECORD TABLE: DIM(631) . . . . .	468
3-16-6	Команда указания расположения записи SET RECORD LOCATION: SETR(635) . . . . .	470
3-16-7	Команда чтения расположения записи GET RECORD NUMBER: GETR(636) . . . . .	472
3-16-8	Команда поиска данных DATA SEARCH: SRCH(181) . . . . .	474
3-16-9	Команда перестановки байтов SWAP BYTES: SWAP(637) . . . . .	477
3-16-10	Команда нахождения максимума FIND MAXIMUM: MAX(182) . . . . .	479
3-16-11	Команда нахождения минимума FIND MINIMUM: MIN(183) . . . . .	482
3-16-12	Команда вычисления суммы SUM: SUM(184) . . . . .	485
3-16-13	Команда вычисления контрольной суммы кадра FRAME CHECKSUM: FCS(180) . . . . .	488
3-17	Команды управления данными . . . . .	492
3-17-1	Команда PID-регулирования PID CONTROL: PID(190) . . . . .	492
3-17-2	Команда контроля пределов выходных данных LIMIT CONTROL: LMT(680) . . . . .	502
3-17-3	Команда контроля пределов зоны нечувствительности DEAD BAND CONTROL: BAND(681) . . . . .	504
3-17-4	Команда контроля зоны нечувствительности DEAD ZONE CONTROL: ZONE(682) . . . . .	506
3-17-5	Команда преобразования данных SCALING: SLC(194) . . . . .	509
3-17-6	Команда преобразования данных SCALING: SLC2(486) . . . . .	513
3-17-7	Команда преобразования данных SCALING: SLC3(487) . . . . .	516
3-17-8	Команда вычисления среднего значения AVERAGE: AVG(195) . . . . .	520
3-18	Подпрограммы . . . . .	524
3-18-1	Команда вызова подпрограммы SUBROUTINE CALL: SBS(091) . . . . .	524
3-18-2	Команда MACRO: MCRO(099) . . . . .	529
3-18-3	Команда начала выполнения подпрограммы SUBROUTINE ENTRY: SBN(092) . . . . .	533
3-18-4	Команда окончания подпрограммы SUBROUTINE RETURN: RET(093) . . . . .	535
3-19	Команды управления прерываниями . . . . .	537

3-19-1	Команда задания прерывания SET INTERRUPT MASK: MSKS(690)	537
3-19-2	Команда чтения данных управления прерыванием READ INTERRUPT MASK: MSKR(692)	540
3-19-3	Команда сброса прерывания CLEARINTERRUPT: CLI(691)	542
3-19-4	Команда блокирования прерываний DISABLE INTERRUPTS: DI(693)	545
3-19-5	Команда разрешения выполнения прерываний ENABLE INTERRUPTS: EI(694)	546
3-19-6	Резюме по управлению прерываниями	547
3-20	Шаговые команды	552
3-20-1	Команды шаговых программ STEP DEFINE and STEP START: STEP(008)/ SNXT(009)	553
3-21	Команды для базовых модулей ввода/вывода	569
3-21-1	Команда регенерации ввода/вывода I/O REFRESH: IORF(097)	569
3-21-2	Команда преобразования в код для семисегментного дисплея 7-SEGMENT DECODER: SDEC(078)	571
3-21-3	Команда чтения памяти ввода/вывода INTELLIGENT I/O READ: IORD(222)	574
3-21-4	Команда записи в память ввода/вывода INTELLIGENT I/O READ: IOWR(223)	577
3-22	Команды последовательного коммуникационного обмена	581
3-22-1	Последовательный коммуникационный обмен	581
3-22-2	Команда выполнения макро протокола PROTOKOL MACRO: PMCR(260)	582
3-22-3	Команда выполнения передачи данных Transmit: TXD(236)	589
3-22-4	Команда выполнения приема данных RECEIVE: RXD(235)	593
3-22-5	Команда изменения установок последовательного порта CHANGE SERIAL PORT SETUP: STUP(237)	598
3-23	Сетевые команды	602
3-23-2	Команда передачи данных при сетевом обмене NETWORK SEND: SEND(090)	606
3-23-3	Команда приема данных при сетевом обмене NETWORK RECEIVE: RECV(098)	611
3-23-4	Передача команды DELIVER COMMAND: CMND(490)	614
3-24	Команды выполнения операций с памятью файлов	621
3-24-1	Команда чтения данных файла READ DATA FILE: FREAD(700)	621
3-24-2	Команда записи данных в файл WRITE DATA FILE: FWRT(701)	627
3-25	Команды дисплея: DISPLAY MESSAGE: MSG(046)	635
3-26	Команды управления часами	638
3-26-1	Команда установки календаря CALENDAR ADD: CADD(730)	638
3-26-2	Команда установки календаря CALENDAR SUBSTRACT: CSUB(731)	641
3-26-3	Команда преобразования данных времени из часов в секунды HOURS TO SECONDS: SEC(065)	644
3-26-4	Команда преобразования данных времени из секунд в часы SECONDS TO HOURS: HMS(066)	646
3-26-5	Команда установки даты CLOCK ADJUSTMENT: DATE(735)	648
3-27	Команды поиска и устранения неисправностей	652
3-27-1	Команда Trace Memory Sampling: TRSM(045)	652
3-28	Команды диагностики отказов	656
3-28-1	Команда допускаемой ошибки FAILURE ALARM: FAL(006)	656
3-28-2	Команда критической ошибки SEVERE FAILURE ALARM: FALS(007)	659
3-28-3	Команда определения точки появления ошибки FAILURE POINT DETECTION: FPD(269)	662
3-29	Прочие команды	671
3-29-1	Команда установки флага переноса SET CARRY: STC(040)	671
3-29-2	Команда очистки (сброса) флага переноса CLEAR CARRY: CLC(041)	671
3-29-3	Команда выбора EM банка SELECT EM BANK: EMBC(281)	672
3-29-4	Команда увеличения максимальной длительности цикла EXTEND MAXIMUM CYCLE TIME: WDT(094)	674
3-30	Команды программирования блоков	677
3-30-1	Введение	677
3-30-2	Команда начала/окончания блочной программы BLOCK PROGRAM BEGIN/END: BPRG(096)/BEND(801)	681
3-30-3	Команда приостановки/повторного запуска блочной программы BLOCK PROGRAM PAUSE/RESTART: BPPS(811)/BPRS(812)	684

3-30-4	Команды ветвления IF(802), ELSE(803) и IEND(804).	686
3-30-5	Команда выхода из условного блока CONDITIONAL BLOCK EXIT(NOT): EXIT(NOT)(806).	691
3-30-6	Команда прерывания выполнения программы на один цикл ONE CYCLE AND WAIT(NOT): WAIT(805)/WAIT(805)NOT.	693
3-30-7	Команда ожидания таймера TIMW(813).	697
3-30-8	Команда ожидания счетчика COUNTER WAIT: CNTW(814)	700
3-30-9	Команда ожидания высокоскоростного таймера HIGH SPEED TIMER WAIT: TMHW(815)	703
3-30-10	Команды управления циклом LOOP(809)/LEND(810)/LEND(810) NOT	705
3-31	Команды обработки текстовой строки	709
3-31-1	Обзор операций управления текстовой строкой	709
3-31-2	Команда перемещения строки MOV STRING:MOV\$(664).	710
3-31-3	Команда соединения строк CONCATENATE STRING: +\$(656)	712
3-31-4	Команда выборки данных из строки GET STRING LEFT: LEFT\$(652)	714
3-31-5	Команда выборки данных из строки GET STRING RIGHT: RIGHT\$(653)	716
3-31-6	Команда выборки данных из середины строки GET STRING MIDDLE: MID\$(654)	719
3-31-7	Команда нахождения заданной текстовой строки в текстовой строке FIND IN STRING: FIND\$(660)	721
3-31-8	Команда вычисления длины текстовой строки STRING LENGTH: LEN\$(650).	723
3-31-9	Команда замещения текстовой строки указанной текстовой строкой REPLACE IN STRING: RPLC\$(661)	725
3-31-10	Команда удаления текстовой строки DELETE STRING: DEL\$(658)	727
3-31-11	Команда обмена содержания текстовых строк EXCHANGE STRING: XCHG\$(665)	730
3-31-12	Команда очистки содержания текстовой строки CLEAR STRING: CLR\$(666)	732
3-31-13	Команда ввода текстовой строки INSERT INTO STRING: INSS\$(657)	733
3-31-14	Команды сравнения строк (670...675)	736
3-32	Команды управления задачами	741
3-32-1	Команда разрешения выполнения задачи TASK ON: TKON(820)	741
3-32-2	Команда перевода задачи в резерв TASK OFF: TKOF(821)	744

## Глава 4

Время выполнения команд и количество шагов	747
--	-----

---

## **Меры предосторожности.**

---

*Раздел содержит описание общих мер предосторожности при эксплуатации Программируемых контроллеров серии CSI и связанного с ними оборудования. Информация, содержащаяся в настоящем разделе, является очень важной для безопасного и надежного использования Программируемых контроллеров. Перед попыткой изменения установок, запуском или эксплуатацией системы Программируемых контроллеров, Вы обязаны внимательно изучить содержание настоящего раздела и понять излагаемую в нем информацию.*

## 1. Категории персонала

Настоящее руководство предназначено для перечисляемого ниже персонала, который также обязан знать электрические системы (инженеры-электрики или равные им по образованию).

- Для персонала, ответственного за установку систем промышленной автоматизации.
- Для персонала, ответственного за разработку систем промышленной автоматизации.
- Для персонала, ответственного за эксплуатацию систем промышленной автоматизации.

## 2. Общие меры предосторожности

Пользователь обязан эксплуатировать оборудование в соответствии с характеристиками, описанными в настоящем руководстве.

Перед использованием изделия в условиях, описание которых не дается в настоящем руководстве, а также в случаях применения оборудования в системах, где существует возможность возникновения серьезной опасности для жизни и собственности, консультируйтесь с местными представителями фирмы OMRON. К таким случаям относятся применение оборудования в системах регулирования ядерных процессов, системах регулирования на железнодорожном транспорте, авиационных системах, на транспорте, в системах управления горением, в медицинском оборудовании, оборудовании для аттракционов, спасательном оборудовании (оборудовании для обеспечения безопасности) и других системах, механизмах и оборудовании.

Убедитесь в том, что мощность и функциональные характеристики применяемого изделия достаточны для работы в системах, механизмах и оборудовании. Непременно обеспечивайте системы, механизмы и оборудование системами двойной защиты.

Настоящее описание дает информацию о порядке программирования и эксплуатации изделий. Перед включением изделия непременно изучите настоящее Руководство, в процессе эксплуатации постоянно обращайтесь к нему для справок.

**Предупреждение!** *Чрезвычайно важно использовать Программируемые контролеры и их блоки только в целях, для которых они предназначены, и только в заданных условиях, особенно при использовании в процессах, которые могут прямо или косвенно воздействовать на жизнь человека. Перед использованием систем Программируемых контроллеров в вышеупомянутых процессах Вы обязаны консультироваться с представителями фирмы OMRON.*

## 3. Меры предосторожности для обеспечения безопасности

**Предупреждение!** *Блок центрального процессора осуществляет инициализацию вводов/выводов даже в том случае, когда программа остановлена (т.е. в режиме программирования). Перед изменением состояния любой части памяти, распределяемой Модулям ввода/вывода, Специальным Модулям или Модулями шины Центрального процессора, предварительно убедитесь в безопасности выполнения такой операции. Любые изменения данных, распределяемых на Модули, могут привести к непредвиденному изменению состояния нагрузки, подключенной к Модулю. Любая из перечисленных ниже операций может привести к изменению состояния памяти.*

- Передача данных памяти ввода/вывода Блоку центрального процессора с помощью Устройства программирования.
- Изменение ранее установленных состояний в памяти при помощи Устройства программирования.
- Принудительная установка/сброс битов при помощи Устройства программирования.
- Передача Блоку Центрального процессора файлов памяти ввода/вывода из Платы памяти или файлов из ЕМ памяти.
- Передача данных памяти ввода/вывода из Главного компьютера или из другого Программируемого контроллера, подключенного к сети.

**Предупреждение!** *Никогда не предпринимайте попыток отключения любого из блоков при включенном напряжении питания. Это может привести к тяжелому поражению электрическим током.*

**Внимание!** *Никогда не прикасайтесь к любой из клемм при включенном напряжении питания. Это может привести к поражению электрическим током.*



- Внимание!** Не предпринимайте попыток разборки, ремонта или модификации изделий. Любая такая попытка может привести к отказу оборудования, его возгоранию или поражению персонала электрическим током.
- Внимание!** Не прикасайтесь к Блоку питания при подключенном напряжении питания либо непосредственно сразу после отключения напряжения питания. Это может привести к поражению персонала электрическим током.
- Внимание!** Изменения параметров в работающей сети производите только после того, как Вы убедитесь в том, что увеличение длительности выполнения цикла не вызовет неблагоприятного влияния на работу системы. В противном случае входные сигналы могут не читаться.
- Внимание!** Перед передачей программы на удаленный узел или перед изменением содержания области ввода/вывода памяти, убедитесь в безопасности выполнения операции на этом узле. Выполнение таких процедур без проверки безопасности может привести к травмированию персонала.
- Внимание!** Затягивайте винты клеммных колодок Блока питания с усилием, указанным в Руководстве по эксплуатации. Ослабление винтов может стать причиной сгорания оборудования или к сбоям в его работе.

#### 4. Меры предосторожности при размещении

- Внимание!** Эксплуатация систем управления не допускается в следующих условиях размещения:
- В местах, подверженных прямому солнечному освещению;
  - В местах, где температура или относительная влажность окружающего воздуха выходят за установленные пределы;
  - В местах, подверженных конденсации влаги вследствие резкого перепада окружающей температуры;
  - В местах, доступных для проникновения газов, способных вызывать коррозию изделия или самовоспламенение;
  - В местах, подверженных осаджению пыли (особенно металлической) или солей;
  - В местах, где оборудование может подвергаться воздействию влаги, масла или химикатов;
  - В местах, где оборудование может подвергаться вибрации или ударам.
- Внимание!** Условия размещения Системы Программируемых контроллеров могут значительно влиять на долговечность и надежность работы системы. Неудовлетворительные окружающие условия могут привести к сбоям в работе оборудования, отказам и другим непредвиденным проблемам. Убедитесь в том, что условия размещения соответствуют заданным параметрам и будут оставаться в установленных пределах на протяжении всего срока эксплуатации системы.

#### 5. Меры предосторожности при эксплуатации

При использовании Системы Программируемых контроллеров выполняйте следующие ниже меры предосторожности.

- В случае необходимости программирования более чем одной задачи, Вы должны использовать СХ-Программатор (программу, работающую в оболочке Windows). С помощью Пульта программирования может программироваться только одна циклическая задача и задачи прерывания. Тем не менее, Пульт программирования может использоваться для редактирования многозадачных программ, первоначально созданных при помощи СХ-Программатора.
- При использовании Специальных Модулей С200Н для выполнения следующих ниже функций, существуют некоторые ограничения доступа к областям и адресам памяти Блоков Центрального процессора серии SC1.
  - Существуют ограничения, касающиеся передачи данных при помощи Блока Центрального процессора, в случае, когда программируется передача данных внутри Модуля ASCII с использованием команд PC READ, PC WRITE и других команд подобных этим.
  - Существуют ограничения на передачу данных при помощи Блока центрального процессора, касающиеся распределенных битов и спецификаций области DM (области и адреса источников данных и точек назначения).



- Области выхода Мастер - модулей ComproBus/D (CIO 0050 - CIO 0099) перекрывают области битов вывода/ввода (CIO 0000 - CIO 0319). Не применяйте автоматическое распределение ввода/вывода в любой из систем, где распределения адресов системы ComproBus/D перекрывает распределение адресов Модулей Ввода/вывода. Вместо этого используйте Устройство программирования или СХ-Программатор для распределения адресов ввода/вывода устройств ComproBus/D вручную, обеспечивая условия, при которых распределяемые слова и биты, используются только один раз. После этого передавайте результирующую таблицу распределения в Блок Центрального процессора. В случае, когда производится попытка осуществления коммуникационного обмена данными в системе ComproBus/D при одновременном распределении одинаковых адресов устройствам ComproBus/D и Модулям ввода/вывода (что возможно при автоматическом распределении адресов), устройства ComproBus/D и Модули ввода/вывода могут выполнить ошибочную операцию.
- Специальные флаги и биты Модулей связи с Программируемым контроллером (далее Модули PC Link) (CIO 0247 - CIO 0250) перекрывают область бит ввода/вывода (CIO 0000 - CIO 0319). Не применяйте автоматическое распределение выводов/вводов в любой из систем, где распределение адресов для Модулей PC Link (в тексте опечатка Н.П.) перекрывает адреса Модулей Ввода/вывода. Вместо этого используйте Устройство программирования или СХ-Программатор для распределения адресов Модулей Ввода/вывода вручную, обеспечивая условия, при которых специальные биты и флаги Модулей PC Link не используются. После этого передавайте результирующую таблицу вводов/выводов в Блок Центрального процессора. В случае, когда производится попытка работы при одновременном распределении Модулям Ввода/вывода специальных бит и флагов Модулей PC Link (что возможно при автоматическом распределении адресов), Модули PC Link и Модули ввода/вывода могут выполнить ошибочную операцию.

**Внимание!** *Тщательно соблюдайте меры предосторожности. Нарушение следующих ниже мер предосторожности может привести к тяжелым, возможно смертельным травмам.*

- При монтаже всегда заземляйте изделия и блоки к заземлению не ниже класса 3 (сопротивление заземления - менее 100ом). Нарушение данной меры предосторожности может привести к поражению персонала электрическим током.
- Заземление класса 3 (сопротивление заземления - менее 100ом) должно подключаться при замыкании клемм GR и LG на блоке питания
- Перед выполнением любой из следующих ниже операций обязательно отключайте напряжение питания Программируемого контроллера. Выполнение любой из перечисленных ниже операций с включенным напряжением питания может привести к поражению электрическим током или сбою в работе оборудования.
  - Монтаж или демонтаж Модулей ввода/вывода, Модулей центрального процессора, Встроенных плат или любых других блоков.
  - Сборка любого из Модулей (имеется в виду установка плат и др. Н.П.).
  - Изменение положений двухпозиционных и многопозиционных переключателей.
  - Подключение кабелей или проводов.
  - Подключение или отключение разъемов.

**Внимание!** *Нарушение следующих ниже мер предосторожности может привести к сбоям в работе Программируемых контроллеров или всей системы, а также к повреждению Программируемого контроллера или его блоков. Непременнo выполняйте эти меры предосторожности.*

- Всегда затягивайте винты клемм, предназначенных для подключения шин питания переменного тока, с усилием, указанным в настоящем Руководстве. Ослабление крепления линий питания может привести к короткому замыканию, сбоям в работе или повреждению оборудования.
- При первоначальном запуске в эксплуатацию Модуля Центрального процессора установите батарею CS1W-BAT1, поставляемую с Модулем. Перед запуском программы произведите при помощи Устройства программирования очистку областей памяти.
- Для использования встроенных часов после установки батареи включите питание и установите часы при помощи Устройства программирования или используя команду DATE (735). До установки времени часы не запускаются.
- В процессе создания файла AUTOEXEC.IOM с помощью Устройства программирования (Пультa программирования или СХ-Программатора), для автоматической передачи данных при запуске установите начальный адрес записи D20000 и убедитесь в том, что объем записываемых данных не превышает объема обла-

ти DM. В случае, когда при включении оборудования файл данных читается в Плате памяти, данные записываются в память Модуля Центрального процессора, начиная с адреса D20000, даже если при создании файла AUTOEXEC.IOM задан другой адрес. Кроме того, если объем данных превышает объем области DM (что возможно при использовании СХ-Программатора), излишняя часть данных будет записана в область EM.

- Всегда включайте питание Программируемого контроллера до включения питания исполнительных устройств системы управления. В противном случае в системе управления могут возникнуть кратковременные ошибки вследствие того, что при включении питания Программируемого контроллера выводы Модулей дискретного вывода и других модулей кратковременно переводятся в состояние ON.
- Потребителем должны быть предусмотрены программные меры предосторожности (в файле) для обеспечения безопасности в случае, когда выводы Модулей вывода остаются в состоянии ON, в результате ошибок в работе вследствие отказов реле, транзисторов или других радиоэлементов.
- Потребителем должны быть предусмотрены программные меры предосторожности (в файле) для обеспечения безопасности в случае поступления неправильного сигнала, пропуска сигнала или приема необычного сигнала вызванного обрывом сигнальных линий, кратковременного прерывания подачи питающего напряжения, или в других случаях.
- Блокирующие устройства, схемы ограничения и другие подобные меры защиты внешних цепей (т.е. вне Программируемого контроллера) должны устанавливаться потребителем.
- Не производите отключение напряжения питания Программируемого контроллера в процессе передачи данных. В частности, не выключайте питание при чтении или записи данных в Плате памяти. Кроме того, не извлекайте Плату памяти в том случае, когда индикатор BUSY светится. Для извлечения Платы памяти вначале нажмите выключатель питания платы, затем дождитесь момента, когда индикатор погаснет. После этого извлекайте плату.
- Если бит удержания ввода/вывода (Hold bit) переведен в состояние ON, выводы Программируемого контроллера не могут переключаться в состояние OFF, и сохраняют свое предшествующее состояние при переводе Программируемого контроллера из рабочего режима (RUN) в режим монитора (Monitor) или в режим программирования (Program). Убедитесь в том, что в этом случае внешняя нагрузка (исполнительный механизм) не производит опасных операций. (Когда работа прекращается вследствие критических ошибок, включая ошибки, происходящие с командой FALS (007), все выводы Модулей вывода будут переведены в состояние OFF, и только состояние внутреннего вывода останется неизменным).
- При использовании напряжения питания от 200 до 230В переменного тока непременно удалите металлическую перемычку с клемм выбора питающего напряжения. Изделие будет выведено из строя, если при подаче питания 200-230В перемычка остается на своем месте.
- Используйте напряжение питания, указанное в соответствующих Руководствах по эксплуатации. Применение других питающих напряжений может привести к повреждению блоков.
- Непременно убедитесь в том, что используется соответствующее напряжение питания заданной мощности и частоты. Будьте особенно внимательны в тех местах, где напряжение питания нестабильно. Нестабильность питающего напряжения может стать причиной сбоев в работе оборудования.
- Устанавливайте внешние прерыватели или предпринимайте другие меры защиты внешних цепей от короткого замыкания. Недостаточные меры защиты от короткого замыкания могут стать причиной сгорания оборудования.
- Не подавайте на входы Модулей ввода напряжения, превышающие номинальные величины. Это может привести к сгоранию Модулей.
- К выводам Модулей вывода не подключайте напряжения, превышающие номинальные величины, или нагрузки, требующие большой мощности переключения, превышающей допустимую мощность. Это может привести к сгоранию Модулей.

- При проведении испытаний прочности изоляции\*\*\*\* отключайте клемму рабочего заземления. Выполнение такого теста при подключенном заземлении может привести к сгоранию оборудования.
- Производите установку Модулей в соответствии с указаниями Руководства по эксплуатации. Нарушение указаний Руководства по установке оборудования может привести к сбоям в работе оборудования.
- Непременно убедитесь в том, что все монтажные винты, винты клеммных колодок и винты кабельных разъемов затянуты с усилием, указанным в соответствующих Руководствах. Не допускайте ослабления винтов, это может привести к сбоям в работе оборудования.
- При монтаже Модулей не удаляйте предохранительные этикетки. Удаление этикеток в процессе монтажа может стать причиной попадания внутрь Модулей посторонних предметов.
- После завершения монтажа оборудования удалите предохранительные этикетки для обеспечения вентиляции Модулей. Оставленные предохранительные этикетки могут стать причиной сбоев в работе оборудования.
- Для подключения проводов к клеммам всегда применяйте обжимаемые контакты. Не подсоединяйте к клеммам проводники с удаленной изоляцией. Это может привести к сгоранию оборудования.
- Не допускайте ошибок при монтаже соединительных линий.
- Перед включением питания всегда дважды проверяйте схему подключения и положения переключателей. Ошибки в подключении Модулей могут привести к сгоранию оборудования.
- Монтаж Модулей производите только после тщательной проверки клеммных блоков и разъемов.
- Убедитесь в том, что клеммные блоки, Блоки памяти, соединительные кабели и другие узлы, снабженные устройствами замыкания, надежно установлены на места и закреплены замками. Ненадежное закрепление таких узлов может стать причиной сбоев в работе оборудования.
- Проверьте положение всех переключателей, содержание области DM памяти, выполните другие приготовления перед началом выполнения операций. Запуск в работу без выполнения необходимых установок или без ввода необходимых данных может привести к непредвиденным действиям оборудования.
- Перед запуском в работу программы пользователя проверьте безопасность ее выполнения. Запуск программы без проведения такой проверки может привести к непредвиденным действиям оборудования.
- Не предпринимайте любых из перечисленных ниже действий до тех пор, пока не убедитесь в том, что эти действия не могут привести к повреждению системы или составляющих ее компонентов.
  - Изменение режима работы Программируемого контроллера.
  - Принудительная установка или сброс любого бита памяти.
  - Изменение ранее установленного значения любого слова или любого значения в памяти.
- Возобновляйте работу только после передачи вновь установленному Модулю Центрального процессора содержания области DM, области HR, и других, необходимых для работы данных. Нарушение этого правила может привести к непредвиденным действиям оборудования.
- Не допускайте натяжения кабелей или их сгибания свыше допустимых пределов. Это может привести к повреждению кабелей.
- Не допускайте расположения тяжелых предметов на кабелях и других соединительных линиях. Это может привести к повреждению кабелей.
- При замене сменных частей убедитесь в соответствии характеристик заменяемых частей. Несоответствие характеристик установленных частей оборудования требуемым значениям может стать причиной сбоев в работе или сгоранию оборудования.
- Перед выполнением любых действий с Модулем непременно прикоснитесь заземленного металлического предмета для снятия электростатического заряда. Не выполнение такой меры предосторожности может привести к сбою в работе или повреждению Модуля.

- При транспортировании или хранении печатных плат накрывайте их антистатическим материалом для защиты от статического электричества. Поддерживайте необходимую температуру хранения плат печатного монтажа.
- При работе с платами печатного монтажа используйте средства защиты рук, т.к. на платах могут быть острые концы проводников или острые концы установленных элементов. Неправильное обращение с платами может привести к травмам.
- Не допускайте короткого замыкания клемм батареи и не производите ее подзарядку, не разбирайте, не нагревайте батарею, не допускайте воздействия открытого огня. Не подвергайте батарею сильным ударам. Все перечисленные действия могут привести к протеканию батареи, разрыву ее корпуса, нагреванию или возгоранию батареи. Удаляйте любую из батарей, которая упала на пол или подверглась сильному удару. Батареи, подвергнутые сильному удару, могут протекать в процессе эксплуатации.
- Стандарты UL требуют, чтобы замена батарей осуществлялась только квалифицированным персоналом. Не допускайте неквалифицированный персонал к выполнению операции замены батарей.

## 6. Соответствие Директивам ЕС

### 6-1 Применяемые Директивы

- Директивы EMC
- Директивы, касающиеся низковольтного оборудования

### 6-2 Общие представления

#### Директивы EMC

Продукция корпорации OMRON, соответствующая требованиям Директив ЕС, также подчиняется соответствующим требованиям стандартов EMC, поэтому изделия могут легко встраиваться в другие устройства или в общий механизм. Описываемая продукция проверена на соответствие требованиям стандартов EMC (см. следующее ниже примечание). В случае, когда изделия используются в системах, подчиняющихся другим стандартам, приведение продукции в соответствие применяемым стандартам должно производиться пользователем самостоятельно.

Характеристики продукции OMRON, подчиняющиеся требованиям ЕС Директив и одновременно относящиеся к требованиям EMC, могут зависеть от конфигурации изделия, способов монтажа, а также используемого оборудования и панелей управления, на которых монтируется оборудование. Вследствие этого потребитель обязан выполнить необходимые проверки окончательно смонтированного оборудования на предмет соответствия стандартам EMC.

**Примечание:** Применяемыми стандартами EMC (стандартами на электромагнитную совместимость) являются:

*EMS (Электромагнитная совместимость): EN61131-2.*

*EMI (Электромагнитные помехи): EN50081-2.*

*(Электромагнитное излучение: на расстоянии 10 м).*

#### Директивы, касающиеся низковольтного оборудования

При эксплуатации низковольтного оборудования, работающего при напряжениях от 50 до 1000В постоянного тока и от 75 до 1500В переменного тока, непременно убедитесь в соответствии этого оборудования стандартам по безопасности для Программируемых контроллеров (EN61131-2).

### 6-3 Соответствие Директивам ЕС

Программируемые контроллеры серии CS1 подчиняются требованиям ЕС Директив. Для обеспечения соответствия требованиям ЕС Директив оборудования, в которое устанавливается Программируемый контроллер серии CS1, контроллер должен устанавливаться следующим образом:

- 1,2,3...** 1. Программируемый контроллер серии CS1 должен устанавливаться внутри панели управления.
2. Для источников постоянного тока, используемых в качестве коммуникационного источника питания и питания вводов/выводов, должна применяться усиленная или двойная изоляция.



3. Программируемые контроллеры серии CS1, подчиняющиеся требованиям ЕС Директив, также подчиняются требованиям Стандарта на общее излучение (EN50081-2). Характеристики электромагнитного излучения (на расстоянии 10 м) могут зависеть от конструкции используемой панели управления, от устройств, подключенных к панели, способа монтажа и других условий. Вследствие этого Вам необходимо убедиться в том, что законченная система соответствует требованиям ЕС Директив.

#### 6-4 Методы снижения помех на релейном выходе

Программируемые контроллеры серии CS1 подчиняются требованиям Стандарта на общее излучение (EN50081-2) EMC Директив. Тем не менее, величина помехи, производимой при переключении выходного реле, может не соответствовать этим стандартам. В этом случае со стороны нагрузки должен подключаться фильтр для подавления помех, либо для снижения помехи к Программируемому контроллеру должны применяться другие подходящие меры.

Меры, предпринимаемые для снижения помех, зависят от устройства, используемого в качестве нагрузки, способа монтажа, конфигурации механизма и т.д. Ниже следуют примеры применяемых мер по снижению помех.

#### Меры по снижению уровня помех

(Для детального ознакомления обратитесь к стандарту EN50081-2).

В случае, когда частота переключения нагрузки в системе, содержащей контроллер, не превышает 5 раз в минуту, применения дополнительных мер по снижению помех не требуется.

В случае, когда частота переключения нагрузки в системе, содержащей контроллер, превышает 5 раз в минуту, требуется применение дополнительных мер по снижению помех.

#### Примеры использования мер по снижению помех

##### Использование цепи RC.

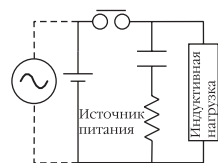


Схема работает как с переменным, так и постоянным током.

##### Характеристики

Если в качестве нагрузки используется реле или соленоид, в системе существует запаздывание между моментом разрыва цепи и моментом переустановки нагрузки.

Если используется напряжение питания 24 или 48В, подключите параллельно нагрузке цепь защиты от броска напряжения. Если напряжение питания от 100 до 200В, подключите цепь защиты между контактами.

##### Требуемые элементы

Емкость конденсатора должна быть от 1 до 0.5 мкФ на каждый 1А тока, протекающего через контакты. Сопротивление резистора - от 0.5 до 1 Ом на каждый 1В напряжения между контактами. Тем не менее, эти значения зависят от нагрузки и характеристик реле. Правильность выбора необходимо определить экспериментальным путем, принимая во внимание тот факт, что конденсатор погашает искрение при разомкнутых контактах, а резистор ограничивает ток, протекающий через нагрузку при замыкании контактов.

Допустимое напряжение конденсатора должно составлять от 200 до 300В. Если конденсатор устанавливается в цепи переменного тока, используйте неполярный конденсатор

##### Использования диода.

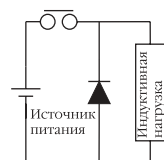


Схема работает как с постоянным током.

##### Характеристики

Диод подключается параллельно нагрузке и превращает энергию, запасаемую в катушке в ток, который протекает через катушку. Таким образом, этот ток превращается в тепло, рассеиваемое на сопротивлении

катушки. Запаздывание между моментом разрыва цепи и моментом переустановки нагрузки в данном случае более длительное, чем в случае использования RC цепи.

#### Требуемые элементы

Величина допустимого обратного напряжения диода должна, по меньшей мере, в 10 раз превышать напряжение в цепи. Величина допустимого тока диода должна быть не меньше тока в нагрузке.

Величина допустимого обратного напряжения диода должна, по меньшей мере, в 3 раза превышать напряжение в цепи, если в цепи с малым напряжением применяется защита от бросков напряжения.

#### Использования варистора.

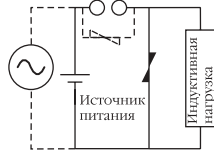


Схема работает как с переменным, так и постоянным током.

#### Характеристики

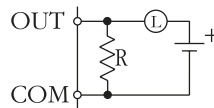
Применение варистора предотвращает возможность приложения высокого напряжения между контактами, благодаря постоянству напряжения варистора. В системе существует запаздывание между моментом разрыва цепи и моментом переустановки нагрузки.

Если используется напряжение питания 24 или 48В, подключите варистор параллельно нагрузке. Если напряжение питания от 100 до 200В, подключите варистор между контактами.

При переключении нагрузки, характеризующейся значительными бросками тока, например, лампы накаливания, примите меры по снижению бросков тока, как показано на следующем рисунке.

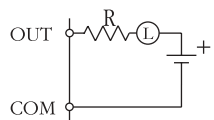
#### Пример 1.

Ток потушенной лампы примерно втрое ниже номинального значения тока лампы накаливания.



#### Пример 2

Включение ограничивающего резистора.







---

## **Глава 1**

### **Введение**

---

*В настоящей главе приводятся основные характеристики команд, а также описание ошибок, которые могут возникать при выполнении команды.*

## 1-1 Основные характеристики команд

### 1-1-1 Объем программы

Параметр объем программы указывает размер области, занимаемой программой пользователя в Модуле центрального процессора, и выражается в количестве шагов программы. Количество шагов, требуемое в области программы пользователя для каждой из команд контроллеров серии CS1, изменяется от 1 до 7 в зависимости от команды и используемого в этой команде операнда.

В следующей ниже таблице показано максимальное количество шагов, которое может программироваться в каждом из Модулей центрального процессора серии CS1.

Модуль центрального процессора	Максимальный объем программы	Количество точек ввода/вывода
CS1H-CPU67-E	250 000 шагов	5120
CS1H-CPU66-E	120 000 шагов	
CS1H-CPU65-E	60 000 шагов	
CS1H-CPU64-E	30 000 шагов	
CS1H-CPU63-E	20 000 шагов	
CS1H-CPU45-E	60 000 шагов	
CS1H-CPU44-E	30 000 шагов	1280
CS1H-CPU43-E	20 000 шагов	960
CS1H-CPU42-E	10 000 шагов	

**Примечание:** Объем программы Программируемых контроллеров серии CS1 измеряется в количестве шагов, в то время как объем программы для Программируемых контроллеров серии С и серии CV, измерялся в количестве слов. В основном, один шаг эквивалентен одному слову. Тем не менее, требуемый объем памяти отличается для некоторых команд контроллеров серии CS1. Поэтому, при преобразовании программ, разработанных для контроллеров других серий, в программу для работы с контроллером серии CS1, объем памяти, рассчитанный из соотношения одно слово = один шаг, приведет к ошибочному результату. При выполнении подобных преобразований обратитесь к информации, изложенной в конце Главы 4 «Время выполнение команд и шаги».

Количество шагов в программе не равно количеству команд. Например, команды LD и OUR требуют одного шага, а команда MOV(021) – трех шагов. Некоторые другие команды требуют выполнения 7-ми шагов. Количество шагов при выполнении команды увеличивается на 1 для каждого из операндов двойной длины. Например, команда MOVL(498) обычно требует выполнения 3-х шагов, однако при указании в качестве исходного слова операнда константы S, команда требует выполнения 4-х шагов. Для ознакомления с количеством шагов, требуемых для выполнения каждой из команд, обратитесь к Главе 4 «Время выполнение команд и шаги».

### 1-1-2 Дифференцированные команды

Большинство команд Программируемых контроллеров серии CS1 обладают как недифференцированными версиями, так и модификациями, дифференцированными по возрастанию (вверх), в то время как некоторые команды имеют также модификации, дифференцированные по убыванию (вниз).

- Недифференцированные команды выполняются каждый раз при наступлении очередности их выполнения.
- Команды, дифференцированные по возрастанию, выполняются только один раз при переходе условия выполнения из состояния OFF в состояние ON.
- Команды, дифференцированные по убыванию, выполняются только один раз при переходе условия выполнения из состояния ON в состояние OFF.

Модификация			
	Тип команды	Действие	Пример
Недифференцированные			
	Команды вывода (команды, требующие условия выполнения)	Команда выполняется в каждом из циклов, когда условие выполнения находится в состоянии ON.	MOV

Модификация		
Тип команды	Действие	Пример
Команды ввода (используемые в качестве условия выполнения)	Обработка бита (например, чтение, сравнение, или проверка) выполняется в каждом из циклов. Условие выполнения является истинным, когда результатом выполнения команды является состояние ON.	LD
Дифференцированные по возрастанию (с префиксом @)		
Команды вывода (команды, требующие условия выполнения)	Команда выполняется только один раз, когда условие выполнения переводится из состояния OFF в состояние ON.	@MOV Команда MOV (021) выполняется только один раз при изменении состояния CIO 000102 из OFF в ON.
Команды ввода (используемые в качестве условий выполнения)	Обработка бита (например, чтение, сравнение, или проверка) выполняется в каждом из циклов. Условие выполнения является истинным только в текущем цикле, когда результатом выполнения команды является переход из состояния OFF в состояние ON	@LD Истинность условия выполнения устанавливается только на время выполнения текущего цикла при изменении состояния CIO 000103 из OFF в ON.
Дифференцированные по убыванию (с префиксом %)		
Команды вывода (команды, требующие условия выполнения)	Команда выполняется только один раз, когда условие выполнения переводится из состояния ON в состояние OFF.	%SET Команда SET выполняется только один раз для каждого изменения состояния CIO 000102 из ON в OFF.
Команды ввода (используемые в качестве условий выполнения)	Обработка бита (например, чтение, сравнение, или проверка) выполняется в каждом из циклов. Условие выполнения является истинным только в текущем цикле, когда результатом выполнения команды является переход из состояния ON в состояние OFF.	%LD Истинность условия выполнения устанавливается только на время выполнения текущего цикла при изменении состояния CIO 000103 из ON в OFF.

**Примечание:** Модификации команд, дифференцированных по убыванию (%), существуют только для команд LD, AND, OR и RSET. Для создания дифференцированных по убыванию модификаций других команд, управляйте выполнением команды с помощью рабочих битов, в свою очередь управляемых командами DIFD (014) или DOWN (522).

### 1-1-3 Модификации команд

Для создания модификаций команд или выполнения немедленной регенерации устанавливайте в командах префиксы (@, % и !).

Модификация		Префикс	Действие
Дифференцирование	Дифференцирование по возрастанию.	@	Создает модификацию команды, дифференцированную по возрастанию.
	Дифференцирование по убыванию.	%	Создает модификацию команды, дифференцированную по убыванию.
Немедленная регенерация		!	При выполнении данной команды осуществляется регенерация данных операнда в Области ввода/вывода или Области специальных модулей.

! @ MOV

мнемоническое изображение команды  
модификация команды, дифференцированной по возрастанию  
модификация для выполнения немедленной регенерации

### 1-1-4 Расположение команды и условия выполнения

В следующей ниже таблице показаны места расположения, куда могут вводиться команды при программировании. В таблице также указано, когда команда требует условия выполнения, а когда не требует. Для ознакомления с подробностями обратитесь к разделу 2 «Сводная таблица команд».

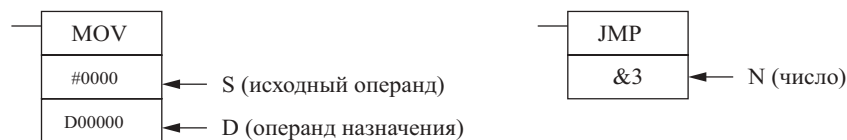
Тип команды		Расположение	Условие выполнения	Примеры
Ввод	Команды, запускающие логические условия.	Возле левой черты или в начале блока команд.	Не требуется.	Команды LD, LD, TST и команды сравнения, такие как LD>.
	Команды соединения	Между командой запуска и командой вывода.	Требуется.	Команды AND, OR, AND TST, команды сравнения, такие как AND>, UP, DOWN, NOT.
Вывод		Возле правой черты.	Требуется	Большинство команд (например OUT и MOV).
			Не требуется	Команды END, JME, FOR, ILC.

В дополнение к данным командам Программируемые контроллеры серии CS1 снабжены командами программирования блоков. Для ознакомления с подробностями обратитесь к описанию команд программирования блоков.

**Примечание:** Если перед командой, требующей условия выполнения, не ставится условие выполнения, определяется ошибка команды при проверке программы из Периферийного устройства.

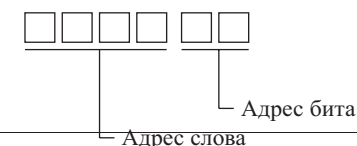
### 1-1-5 Ввод данных в операнды

Операнды являются параметрами, которые задаются предварительно, вместе с адресами ввода/вывода или константами, используемыми при выполнении команд. Существует три типа операндов: исходные операнды, операнды назначения и числа.

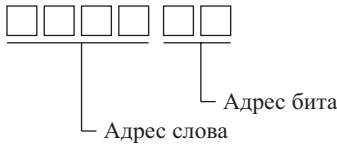
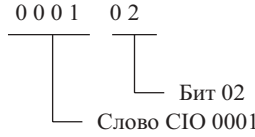


Операнд	Обычный код	Содержание		
		Исходный	Адрес, содержащий данные, или собственно данные.	S
		C	Данные управления	Данные управления с битом или битами, управляющими выполнением команды.
Операнд назначения	Адрес, где будут сохраняться данные.	D	—	
Номер	Содержит число, например номер перехода или номер подпрограммы.	N	—	

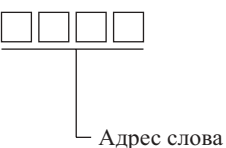
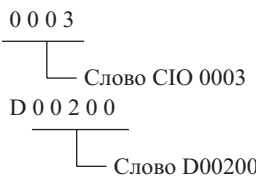
**Примечание:** Операнды команд могут упоминаться согласно их расположению в команде (т.е. первый операнд, второй операнд и т.д.). Коды, применяемые для операндов, могут изменяться в зависимости от функционального назначения операнда.



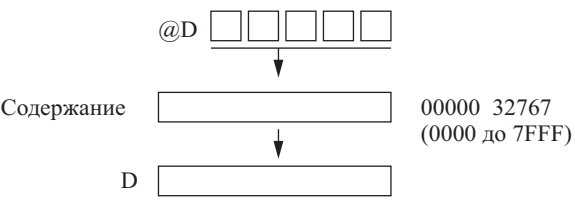
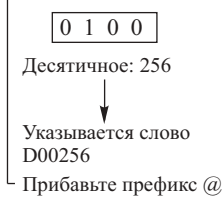
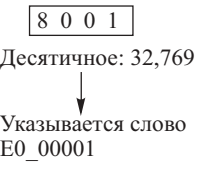
**Указание адресов битов**

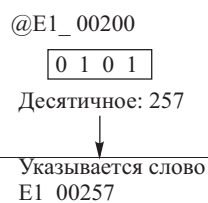
Описание	Пример	Пример команды
<p>Для указания адреса бита, непосредственно укажите адрес слова и бита.</p>  <p>Примечание: Формат адрес слова + адрес бита не применяется для флагов завершения таймеров/ счетчиков или для флагов задач.</p>		LD 000102

**Указание адресов слов**

Описание	Пример	Пример команды
<p>Для указания адреса слова, непосредственно укажите адрес слова.</p> 		MOV 0003 D00200

**Косвенная адресация DM/EM памяти в двоичном коде**

Описание	Пример	Пример команды
<p>Когда перед адресом DM или EM памяти устанавливается префикс @, содержание этого слова указывает на другое слово, используемое в качестве операнда. Содержание слова изменяется от 0000 до 7FFF (0...32767) в соответствии с адресом желаемого слова в Области DM или EM.</p> 	—	—
<p>Когда содержание @D...находится в пределах 0000 до 7FFF (00000...32767), указывается соответствующее слово в пределах от D00000 до D32767.</p>	<p>@D 00300</p> 	MOV #0001 @D00300
<p>Когда содержание @Eп...находится в пределах 8000 до FFFF (32768...65535), указывается соответствующее слово в пределах от E0_00000 до E0_32767 в банке 0 EM памяти.</p>	<p>@D 00300</p> 	—



Описание	Пример	Пример команды
Когда содержание @Eп...находится в пределах 0000 до 7FFF (00000...32767), указывается соответствующее слово в пределах от E0_00000 до E0_32767	@E1_00200 <div style="text-align: center;"> <span style="border: 1px solid black; padding: 2px;">0 1 0 1</span>                      Десятичное: 257                      ↓                      Указывается слово E1_00257                 </div>	MOV #0001 @ E1_00200
Когда содержание @Eп...находится в пределах 8000 до FFFF (32768...65535), указывается соответствующее слово в пределах от E(+1)_00000 до E(+1)_32767 в следующем банке EM памяти.	@E1_00200 <div style="text-align: center;"> <span style="border: 1px solid black; padding: 2px;">8 0 0 2</span>                      Десятичное: 32770                      ↓                      Указывается слово E2_00002                 </div>	

**Примечание:** В случае, когда в начальных установках Программируемого контроллера выбран двоичный код, адреса области DM и адреса текущего EM банка (банки 0...С) используются как последовательные адреса памяти. Если косвенно указываемое слово DM памяти содержит значение, большее, чем 32767, будет указываться одно слово в банке 0. Например, если при косвенной адресации слово DM памяти содержит шестнадцатеричное число 8000 (32768), указывается слово E00000 в банке 0. Если при косвенной адресации EM памяти слово содержит значение, большее, чем 32767, будет указываться слово в следующем EM банке. Например, если при косвенной адресации слово EM памяти в банке 2 содержит шестнадцатеричное число 8000 (32768), указывается слово E3\_00000.

**Косвенная адресация DM/EM памяти в двоично-десятичном коде**

Описание	Пример	Пример команды
Когда перед DM или EM адресом устанавливается префикс «D», двоично-десятичное значение в этом слове указывает на другое слово, используемое в качестве операнда. Диапазон значений – от 0000 до 9999, соответствующие адресам слов в DM или EM области.  <div style="text-align: center;"> </div>	D 00200 <div style="text-align: center;"> <span style="border: 1px solid black; padding: 2px;">0 1 0 0</span>                      ↓                      Указывается слово D00100                      *Прибавьте префикс *                 </div>	MOV #-0001 D00200

**Адресация индексных регистров**

Метод			
Описание	Пример	Пример команды	
<b>Непосредственная адресация индексных регистров</b>			
Команда MOVR(560) перемещает адрес слова или бита памяти внутреннего ввода/вывода в индексный регистр (IR0...IR15). (Команда MOVRW(561) перемещает адрес текущего значения таймера или счетчика в индексный регистр.)	IR0 IR2	MOVR 0010 IR0 Осуществляет запись слова CIO 0010 памяти внутреннего ввода/вывода в регистр IR0. MOVR 000102 IR2 Осуществляет запись слова CIO 000102 памяти внутреннего ввода/вывода в регистр IR2.	
<b>Косвенная адресация с помощью индексных регистров.</b>			
Основные действия (без смещения)	Слово или бит, содержащийся в IR, используется в качестве операнда. Для индикации косвенной адресации перед индексным регистром установите запятую. (Указание слова или бита определяется командой или операндом.)	,IR0 ,IR1	LD ,IR0 Осуществляет запись состояния бита в адрес памяти ввода/вывода, содержащийся в IR0. MOV #0001 ,IR1 Осуществляет перемещение #0001 в адрес памяти ввода/вывода, содержащийся в IR1.

Метод		Описание	Пример	Пример команды
	Постоянное смещение	Значение смещения (-2048...+2047) прибавляется к адресу памяти ввода/вывода, содержащемуся в IR и результирующий адрес используется в качестве операнда.  (Смещение преобразуется в двоичное значение при выполнении команды.)	+5 ,IR0 +31,IR1	LD +5 ,IR0 Осуществляет прибавление 5 к адресу памяти ввода/вывода, содержащемуся в IR0, и записывает состояние бита в этот адрес.  MOV #0001 +31 , IR1 Осуществляет прибавление 31 к адресу памяти ввода/вывода, содержащемуся в IR1, и перемещает #0001 в слово в этом адресе.
	Смещение DR	Содержание Регистра данных в двоичном коде со знаком суммируется с адресом памяти ввода/вывода, содержащимся в IR, после этого результирующий адрес используется в качестве операнда.	DR0 ,IR0 DR0 ,IR1	LD DR0 , IR0 Осуществляет прибавление содержания DR0 к адресу памяти ввода/вывода, содержащемуся в IR0, после этого записывает состояние бита в этот адрес.  MOV #0001 DR0 , IR1 Осуществляет прибавление содержания DR0 к адресу памяти ввода/вывода, содержащемуся в IR1, и перемещает #0001 в слово в этом адресе.
	Автоматическое увеличение	После прочтения адреса памяти ввода/вывода в регистре IR содержание индексного регистра увеличивается на 1 или 2.  Увеличение на 1: ,IR+ Увеличение на 2: ,IR++  Примечание: При выполнении команды индексные регистры увеличиваются, даже если возникает ошибка и флаг ошибки переводится в состояние ON.	,IR0+ + ,IR1+	LD , IR0 + + Записывает состояние бита в адрес памяти ввода/вывода, содержащийся в IR0, затем увеличивает регистр на 2.  MOV #0001 ,IR1+ Осуществляет перемещение #0001 в слово в адресе памяти ввода/вывода, содержащемуся в IR1, затем увеличивает регистр на 1.
	Автоматическое уменьшение	Содержание индексного регистра IR уменьшается на 1 или 2, после этого адрес памяти ввода/вывода в регистре используется в качестве операнда.  Уменьшение на 1: , - IR Уменьшение на 2: , - - IR  Примечание: При выполнении команды индексные регистры уменьшаются, даже если возникает ошибка и флаг ошибки переводится в состояние ON.	,IR0+ + ,IR1+	LD , - - IR0 Уменьшает содержание регистра IR0 на 2 и записывает состояние бита в этот адрес памяти ввода/вывода.  MOV #0001 , - IR1 Уменьшает содержание регистра IR0 на 1, затем осуществляет перемещение #0001 в слово в адресе памяти ввода/вывода.

**Примечание:** Убедитесь в том, что содержание индексного регистра обозначает действительный адрес памяти ввода/вывода.

#### Указание констант

Метод	Применяемые операнды	Формат данных	Код	Диапазон значений
Константа (постоянная, данные длиной 16 битов)	Все данные в двоичном коде и двоичные данные в диапазоне	Двоичные данные без знака	#	#0000...#FFFF
		Десятичные данные со знаком	+ -	-2147483648...+2147483647
		Десятичные данные без знака	&	&0...&66535
	Все данные в двоично-десятичном коде и двоично-десятичные данные в диапазоне	Двоично-десятичные данные	#	#0000...#9999
Константа (постоянная, данные длиной 32 бита)	Все данные в двоичном коде и двоичные данные в диапазоне	Двоичные данные без знака	#	#0000 0000...FFFF FFFF
		Десятичные данные со знаком	+ -	-2147483648...+2147483647
		Десятичные данные без знака	&	&0...&4294967295
	Все данные в двоично-десятичном коде и двоично-десятичные данные в диапазоне	Двоично-десятичные данные	#	#0000 0000...#9999 9999

**Примечание:** Формат десятичных данных без знака поддерживается только СХ-программатором.



**Задание текстовой строки**

Текст записывается в ASCII коде (один байт/один знак), начиная с байта младшего разряда и младшего слова в ряду.

Если в строке содержится нечетное число символов, в старшем байте последнего слова записывается значение 00 (ноль).

Если в строке содержится четное число символов, в старшем байте последнего слова записывается значение 0000 (два нуля).

В следующей ниже таблице представлены символы, которые могут выражаться в коде ASCII.

Четыре правых бита	Четыре левых бита										A	B	C	D	E	F
	0	1	2	3	4	5	6	7	8	9						
0			Sp	0	@	P	`	p			Японские иероглифы					
1			!	1	A	Q	a	q								
2			"	2	B	R	b	r								
3			#	3	C	S	c	s								
4			\$	4	D	T	d	t								
5			%	5	E	U	e	u								
6			&	6	F	V	f	v								
7			'	7	G	W	g	w								
8			(	8	H	X	h	x								
9			)	9	I	Y	i	y								
A			*	:	J	Z	j	z								
B			+	;	K	[	k	{								
C			,	<	L		l									
D			-	=	M	]	m	}								
E			.	>	N	^	n	~								
F			/	?	O	_	o									

**1-1-6 Форматы данных**

Ниже представлены форматы данных, которые могут использоваться с Программируемыми контроллерами серии CS1.

**Двоичные данные без знака  
Формат данных**

	15															0				
Двоичные	2 <sup>15</sup>	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>				
Десятичные	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1				
Шестнадцатеричные	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>				

**Значения в десятичном виде**

от 0 до 65535

**4 знака в шестнадцатеричном виде**

от 0000 до FFFF

**Двоичные данные со знаком  
Формат данных**

	15															0				
Двоичные	2 <sup>15</sup>	2 <sup>14</sup>	2 <sup>13</sup>	2 <sup>12</sup>	2 <sup>11</sup>	2 <sup>10</sup>	2 <sup>9</sup>	2 <sup>8</sup>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>				
Десятичные	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1				
Шестнадцатеричные	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>				

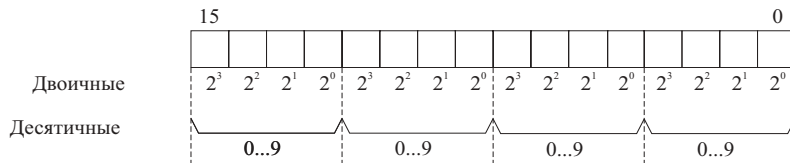
↑  
Бит знака: 0 - положительный, 1 - отрицательный

**Значения в десятичном виде**

от -32768 до +32768

**4 знака в шестнадцатеричном виде**

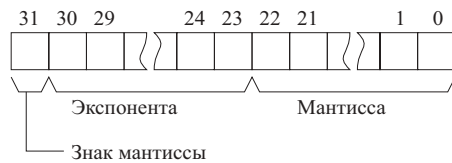
от 8000 до 7FFF

**Двоично-десятичные данные (десятичные в двоичном коде)****Формат данных****Значения в десятичном виде**

от 0 до 9999

**4 знака в шестнадцатеричном виде**

от 0000 до 9999

**Десятичные данные с плавающей запятой****Формат данных**

$$\text{Значение} = (-1)^{\text{Sign}} * 1. [\text{мантисса}] * 2^{\text{Экспонента}}$$

**Двоичные числа со знаком**

Отрицательные числа со знаком выражаются в виде дополнения абсолютного шестнадцатеричного значения по базе 2. Для десятичного значения 12.375 абсолютное значение в шестнадцатеричном коде равно 3039. Дополнение по базе 2 равно 10000 – 3039 (шестн.) или CFC7.

Для преобразования отрицательного двоичного числа (CFC7) в десятичное число, вычислите дополнение по базе 2 (10000 – CFC7 = 3039), преобразуйте в десятичное число (3039 шестн. = 12345 десятичное), и прибавьте знак минуса (- 12345).

**1-2 Проверки выполнения команд****1-2-1 Ошибки, возникающие в процессе выполнения команд**

Проверка операндов команд и их размещение производится при вводе команды из Периферийного устройства, и из Периферийного устройства производится проверка программы, однако эти проверки не являются гарантией отсутствия ошибок. При выполнении команды могут происходить следующие ошибки.

**Ошибка выполнения команды (флаг ER переводится в состояние ON)**

Обычно, ошибки выполнения команды не являются критическими ошибками, однако в начальных установках Программируемого контроллера можно предусмотреть установку, обеспечивающую оценку ошибки выполнения команды как критической ошибки. Если такая установка произведена, при возникновении ошибки выполнения программы флаг ошибки выполнения команды (A29508) переводится в состояние ON и выполнение программы прекращается.

**Ошибка доступа (флаг ARE переводится в состояние ON)**

Обычно, ошибка доступа не является критическими ошибками, однако в начальных установках Программируемого контроллера можно предусмотреть установку, обеспечивающую оценку ошибки доступа как критической ошибки. Если такая установка произведена, при возникновении ошибки доступа флаг ошибки доступа (A29510) и флаг ошибки косвенной адресации памяти DMEM (A29509) переводится в состояние ON и выполнение программы прекращается.

**Ошибка при выполнении недействительной команды**

При возникновении ошибки, обусловленной попыткой выполнения недействительной команды, флаг (A29514) переводится в состояние ON и выполнение программы прерывается.

Ошибка вследствие переполнения памяти пользователя (Памяти программы пользователя)

При возникновении такой ошибки флаг ошибки переполнения памяти программы пользователя переводится в состояние ON и выполнение программы прерывается.

## 1-2-2 Критические ошибки (ошибки программы)

Выполнение программы прерывается при возникновении одной из следующих ниже ошибок программы. В этом случае номер задачи, в процессе выполнения которой остановлена программа, записывается в A294 и адрес программы записывается в A298 A299.

При необходимости используйте эти слова для записи ошибок программы и корректировки программы.

Адрес	Описание
A294	В это слово записывается номер текущей задачи, когда выполнение программы прерывается вследствие возникновения ошибки программы. Циклические задачи имеют номера от 0000 до 001F (циклические задачи 0 31). Задачи прерывания имеют номера от 8000 до 80FF (задачи прерывания 0 255).
A298 и A299	В эти слова записывается адрес текущей программы, когда выполнение программы прерывается вследствие возникновения ошибки программы. A299 содержит цифры старших разрядов адреса программы, а A298 содержит цифры младших разрядов адреса программы.

Все ошибки, при возникновении которых флаг ошибки и флаг ошибки доступа переводятся в состояние ON, принимаются в качестве ошибок программы. В следующей ниже таблице приводится перечень ошибок программы. В начальных установках программируемого контролера можно задать прерывание выполнения программы при возникновении одной из этих ошибок.

Тип ошибки	Описание	Флаги
Отсутствие команды END	В программе отсутствует команда END(001).	Флаг ошибки вследствие отсутствия команды END (A29511).
Ошибка задачи	Существует три возможных причины возникновения ошибки задачи: Отсутствует разрешенная к выполнению циклическая задача. В задаче отсутствует программа. Генерируется прерывание, однако соответствующая задача прерывания отсутствует.	Флаг ошибки задачи (A29512).
Ошибка выполнения команды*	Модуль центрального процессора осуществляет попытку выполнения команды, однако, данные операнда некорректны. *Если в начальных установках программируемого контролера задана установка, определяющая ошибку команды в качестве критической ошибки (ошибки программы), флаг ошибки выполнения команды (A29508) переводится в состояние ON и выполнение программы прерывается.	Флаг ошибки (ER) флаг ошибки выполнения команды (A29508).
Ошибка доступа*	Существует пять возможных причин определения ошибки доступа: Осуществляется чтение/ запись в области параметров. Осуществляется запись в память, которая не установлена. Осуществляется чтение/ запись в банк EM, который является EM памятью файлов. Осуществляется запись в область, предназначенную только для чтения. Содержание слов DM/EM памяти введено не в двоично-десятичном коде, когда Программируемому контроллеру задана косвенная адресация в двоично-десятичном коде. * Если в начальных установках программируемого контролера задана установка, определяющая ошибку команды в качестве критической ошибки (ошибки программы), флаг ошибки вследствие запрещенного доступа (A29510) переводится в состояние ON и выполнение программы прерывается.	Флаг ошибки доступа (AER), флаг ошибки вследствие запрещенного доступа (A29510).
Ошибка косвенной адресации DM/EM памяти	Содержание слов DM/EM памяти введено не в двоично-десятичном коде, когда Программируемому контроллеру задана косвенная адресация в двоично-десятичном коде. *Если в начальных установках программируемого контролера задана установка, определяющая ошибку команды в качестве критической ошибки (ошибки программы), флаг ошибки косвенной адресации DM/EM (A29509) переводится в состояние ON и выполнение программы прерывается.	Флаг ошибки доступа (AER), флаг ошибки косвенной адресации DM/EM (A29509).

---

Тип ошибки	Описание	Флаги
Ошибка переполнения при дифференцировании*	Дифференцированные команды вводились и удалялись в процессе оперативного редактирования более 31072 раз.	Флаг ошибки переполнения при дифференцировании (A29513).
Ошибка переполнения памяти пользователя	Превышен последний адрес памяти программы пользователя.	Флаг ошибки переполнения памяти пользователя (A29515).
Ошибка недействительной команды	Программа содержит команду, которая выполняться не может.	Флаг ошибки недействительной команды (A29514).



---

## **Глава 2**

### **Краткое описание команд**

---

*В настоящем разделе приводится описание команд, используемых Программируемыми контроллерами серии CSI.*



## 2-1 Функциональная классификация команд

В следующей ниже таблице приводится перечень команд контроллеров серии CS1 по функциональному назначению. (Команды представлены в порядке функционального назначения согласно *разделу 3 «Команды»*.)

Классификация	Мнемоника	Команда	Мнемоника	Команда	Мнемоника	Команда
Подкласс						
<b>Основные команды</b>						
Ввод	LD	LOAD	LD NOT	LOAD NOT	AND	AND
	AND NOT	AND NOT	OR	OR	OR NOT	OR NOT
	AND LD	AND LOAD	OR LD	OR LOAD		
Вывод	OUT	OUTPUT	OUT NOT	OUTPUT NOT		
<b>Команды последовательного ввода</b>						
	NOT	NOT	UP	CONDITION ON	DOWN	CONDITION OFF
Битовый тест	LD TST	LD BIT TEST	LD TSTN	LD BIT TEST NOT	AND TST	AND BIT TEST NOT
	AND TSTN	AND BIT TEST NOT	OR TST	OR BIT TEST	OR TSTN	OR BIT TEST NOT
<b>Команды последовательного вывода</b>						
	KEEP	KEEP	DIFU	DIFFERENTIATE UP	DIFD	DIFFERENTIATE DOWN
Установка/сброс	SET	SET	RSET	RESET	SETA	MULTIPLE BIT SET
	RSTA	MULTIPLE BIT RESET	–	–	–	–
<b>Команды последовательного управления</b>						
–	END	END	NOP	NO OPERATION	–	–
Блокирование	IL	INTERLOCK	ILC	INTERLOCK CLEAR	–	–
Переход	JMP	JUMP	JME	JUMP END	CJP	CONDITIONAL JUMP
	CJPN	CONDITIONAL JUMP	JMPO	MULTIPLE JUMP	JMEO	MULTIPLE JUMP END
Повторение	FOR	FOR-NEXT LOOPS	BREAK	BREAK LOOP	NEXT	FOR-NEXT LOOPS
<b>Команды управления таймерами и счетчиками</b>						
Таймер (с номером таймера)	TIM	TIMER	TIMH	HIGH-SPEED TIMER	TMHH	ONE-MS TIMER
	TTIM	ACCUMULATIVE TIMER	–	–	–	–
Таймер (без номера таймера)	TIML	LONG TIMER	MTIM	MULTI-OUTPUT TIMER	–	–
Счетчик (с номером счетчика)	CNT	COUNTER	CNTR	REVERSIBLE COUNTER	CNR	RESET TIMER/COUNTER
<b>Команды сравнения</b>						
Символьное сравнение	LD, AND, OR + =, <, <, J, >, i	Символьное сравнение (без знака)	LD, AND, OR + =, <, <, J, >, i + L	Символьное сравнение (двойное слово, без знака)	LD, AND, OR + =, <, <, J, >, i + S	Символьное сравнение (со знаком)
	LD, AND, OR + =, <, <, J, >, i + SL	Символьное сравнение (двойное слово, со знаком)	–	–	–	–
Сравнение данных (Флаги состояний)	CMP	COMPARE	CMPL	DOUBLE COMPARE	CPS	SIGNED BINARY COMPARE
	CPSL	DOUBLE SIGNED BINARY COMPARE	–	–	–	–

Классификация	Мнемоника	Команда	Мнемоника	Команда	Мнемоника	Команда
Подкласс						
Сравнение таблиц	BCMP	UNSIGNED BLOCK COMPARE	TCMP	TABLE COMPARE	MCMP	MULTIPLE COMPARE
Команды перемещения данных						
Простое/двойное слово	MOV	MOVE	MVN	MOVE NOT	MOVL	DOUBLE MOVE
	MVNL	DOUBLE MOVE NOT	–	–	–	–
Бит/цифра	MOVB	MOVE BIT	MOVD	MOVE DIGIT	–	–
Индексный регистр	MOVR	MOVE REGISTER TO	MOVW	MOVE TIMER/COUNTER PV TO REGISTER	–	–
Обмен	XCHG	DATA EXCHANGE	XCGL	DOUBLE DATA EXCHANGE	–	–
Передача блока/бита	XFER	BLOCK TRANSFER	BSET	BLOCK SET	XFRB	MULTIPLE BIT TRANSFER
Распределение/сбор	COLL	DATA COLLECT	DIST	SINGLE WORD DISTRIBUTE	–	–
Команды смещения данных						
Смещение на 1 бит	SFT	SHIFT REGISTER	SFTR	REVERSIBLE SHIFT REGISTER	ASLL	DOUBLE SHIFT LEFT
	ASL	ARITHMETIC SHIFT LEFT	ASR	ARITHMETIC SHIFT RIGHT	ASRL	DOUBLE SHIFT RIGHT
0000 шестн. Асинхронный	ASFT	ASYNCHRONOUS SHIFT REGISTER	–	–	–	–
Смещение слова	WSFT	WORD SHIFT	–	–	–	–
Вращение 1 бита	ROL	ROTATE LEFT	ROLL	DOUBLE ROTATE LEFT	RLNC	ROTATE LEFT WITHOUT CARRY
	RLNL	DOUBLE ROTATE LEFT WITHOUT CARRY	ROR	ROTATE RIGHT	RORL	DOUBLE ROTATE RIGHT
	RRNC	ROTATE RIGHT WITHOUT CARRY	RRNL	DOUBLE ROTATE RIGHT WITHOUT CARRY	–	–
Смещение на одну цифру	SLD	ONE DIGIT SHIFT LEFT	SRD	ONE DIGIT SHIFT RIGHT	–	–
Смещение n битов данных	NSFL	SHIFT N-BIT DATA LEFT	NSFR	SHIFT N-BIT DATA RIGHT	–	–
Смещение на n-битов	NASL	SHIFT N-BITS LEFT	NSLL	DOUBLE SHIFT N-BITS LEFT	NASR	SHIFT N-BITS RIGHT
	NSRL	DOUBLE SHIFT N-BITS RIGHT	–	–	–	–
Команды увеличения/уменьшения						
BCD (двоично-десятичные)	++B	INCREMENT BCD	++BL	DOUBLE INCREMENT BCD	--B	DECREMENT BCD
	--BL	DOUBLE DECREMENT BCD	–	–	–	–
BIN (двоичные)	++	INCREMENT BINARY	++L	DOUBLE INCREMENT BINARY	--	DECREMENT BINARY
	--L	DOUBLE DECREMENT BINARY	–	–	–	–
Символьные математические команды						
BIN (двоичные числа) сложение	+	SIGNED BINARY ADD WITHOUT CARRY	+L	DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+C	SIGNED BINARY ADD WITH CARRY

Классификация	Мнемоника	Команда	Мнемоника	Команда	Мнемоника	Команда
Подкласс						
	+CL	DOUBLE SIGNED BINARY ADD WITH CARRY	–	–	–	–
BCD (двоично-десятичные числа) сложение	+B	BCD ADD WITHOUT CARRY	+BL	DOUBLE BCD ADD WITHOUT CARRY	+BC	BCD ADD WITH CARRY
	+BCL	DOUBLE BCD ADD WITH CARRY	–	–	–	–
BIN (двоичные числа) вычитание	-	SIGNED BINARY SUBTRACT WITHOUT CARRY	-L	DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	-C	SIGNED BINARY SUBTRACT WITH CARRY
	-CL	DOUBLE SIGNED BINARY WITH CARRY	–	–	–	–
BIN (двоичные числа) умножение	*	SIGNED BINARY MULTIPLY	*L	DOUBLE SIGNED BINARY MULTIPLY	*U	UNSIGNED BINARY MULTIPLY
	*UL	DOUBLE UNSIGNED BINARY MULTIPLY	–	–	–	–
BIN (двоичные числа) деление	/U	UNSIGNED BINARY DIVIDE	/UL	DOUBLE UNSIGNED BINARY DIVIDE	/	SIGNED BINARY DIVIDE
	/L	DOUBLE SIGNED BINARY DIVIDE	–	–	–	–
BCD (двоично-десятичные числа) вычитание	-B	BCD SUBTRACT WITHOUT CARRY	-BL	DOUBLE BCD SUBTRACT WITHOUT CARRY	-BC	BCD SUBTRACT WITH CARRY
	-BCL	DOUBLE BCD SUBTRACT WITH CARRY	–	–	–	–
BCD (двоично-десятичные числа) умножение	*B	BCD MULTIPLY	*BL	DOUBLE BCD MULTIPLY	–	–
BCD (двоично-десятичные числа) деление	/B	BCD DIVIDE	/BL	DOUBLE BCD DIVIDE	–	–
Команды преобразования						
Преобразование двоично-десятичного числа в двоичное число	BIN	BCD-TO-BINARY	BINL	DOUBLE BCD-TO-DOUBLE BINARY	BCD	BINARY-TO-BCD
	BCDL	DOUBLE BINARY-TO-DOUBLE BCD	NEG	2'S COMPLEMENT	NEGL	DOUBLE 2'S COMPLEMENT
	SIGN	16-BIT TO 32-BIT SIGNED BINARY	–	–	–	–
Декодирование/кодирование	MLPX	DATA DECODER	DMPX	DATA ENCODER	–	–
Преобразование кода ASCII в шестнадцатеричный код	ASC	ASCII CONVERT	HEX	ASCII TO HEX	–	–
Преобразование строка/колонка	LINE	COLUMN TO LINE	COLM	LINE TO COLUMN	–	–
Преобразование двоичного числа со знаком в двоично-десятичное число	BINS	SIGNED BCD-TO-BINARY	BISL	DOUBLE SIGNED BCD-TO-BINARY	BCDS	SIGNED BINARY-TO-BCD
	BDSL	DOUBLE SIGNED BINARY-TO-BCD	–	–	–	–

Классификация	Мнемоника	Команда	Мнемоника	Команда	Мнемоника	Команда
Подкласс						
Логические команды						
Дополнение	COM	COMPLEMENT	COML	DOUBLE COMPLEMENT	–	–
Логические операции И/ИЛИ	ANDW	LOGICAL AND	ORW	LOGICAL OR	XORW	EXCLUSIVE OR
	XNRW	EXCLUSIVE NOR	ANDL	DOUBLE LOGICAL AND	ORWL	DOUBLE LOGICAL OR
	XORL	DOUBLE EXCLUSIVE OR	XNRL	DOUBLE EXCLUSIVE NOR	–	–
Специальные математические команды						
–	ROTB	BINARY ROOT	ROOT	BCD SQUARE ROOT	APR	ARITHMETIC PROCESS
	FDIV	FLOATING POINT DIVIDE	BCNT	BIT COUNTER	–	–
Команды математических операций с плавающей запятой						
Преобразование числа с плавающей запятой в двоичное число	FIX	FLOATING 16-BIT TO	FIXL	FLOATING 32-BIT TO	FLT	16-BIT FLOATING TO
	FLTL	32-BIT FLOATING TO	–	–	–	–
Основные математические команды для чисел с плавающей запятой	+F	FLOATINGPOINT ADD	-F	FLOATINGPOINT SUBTRACT	/F	FLOATINGPOINT DIVIDE
	*F	FLOATINGPOINT MULTIPLY	–	–	–	–
Тригонометрические функции для чисел с плавающей запятой	RAD	DEGREES TO RADIANS	DEG	RADIANS TO DEGREES	SIN	SINE
	COS	COSINE	TAN	TANGENT	ASIN	ARC SINE
	ACOS	ARC COSINE	ATAN	ARC TANGENT	–	–
Математические команды для чисел с плавающей запятой	SQRT	SQUARE ROOT	EXP	EXPONENT	LOG	LOGARITHM
	PWR	EXPONENTIAL POWER	–	–	–	–
Работа с таблицами						
Работа со стеком	SSET	SET STACK	PUSH	PUSH ONTO STACK	FIFO	FIRST IN FIRST OUT
	LIFO	LAST IN FIRST OUT	–	–	–	–
Обработка первой записи/нескольких слов	DIM	DIMENSION RECORD TABLE	SETR	SET RECORD LOCATION	GETR	GET RECORD NUMBER
Обработка записи в слово	SRCH	DATA SEARCH	MAX	FIND MAXIMUM	MIN	FIND MINIMUM
	SUM	SUM	FCS	FRAME CHECKSUM	–	–
Обработка байта	SWAP	SWAP BYTES	–	–	–	–
Команды управления данными						
–	PID	PID CONTROL	LMT	LIMIT CONTROL	BAND	DEAD BAND CONTROL
	ZONE	DEAD ZONE CONTROL	SCL	SCALING	SCL2	SCALING 2
	SCL3	SCALING 3	AVG	AVERAGE	–	–
Команды подпрограмм						
–	SBS	SUBROUTINE CALL	SBN	SUBROUTINE ENTRY	RET	SUBROUTINE RETURN
	MCRO	MACRO	–	–	–	–
Команды управления прерываниями						
–	MSKS	SET INTERRUPT MASK	MSKR	READ INTERRUPT MASK	CLI	CLEAR INTERRUPT
	DI	DISABLE INTERRUPTS	EI	ENABLE INTERRUPTS	–	–
Шаговые команды						

Классификация	Мнемо-ника	Команда	Мнемо-ника	Команда	Мнемо-ника	Команда
Подкласс						
–	STEP	STEP DEFINE	SNXT	STEP START	–	–
Команды Базовых модулей						
–	IORF	I/O REFRESH	SDEC	7-SEGMENT DECODER	IORD	INTELLIGENT I/O READ
	IOWR	INTELLIGENT I/O WRITE	–	–	–	–
Команды последовательного коммуникационного обмена						
–	PMCR	PROTOCOL MACRO	TXD	TRANSMIT	RXD	RECEIVE
	STUP	CHANGE SERIAL PORT SETUP	–	–	–	–
Сетевые команды						
–	SEND	NETWORK SEND	RECV	NETWORK RECEIVE	CMND	DELIVER COMMAND
Команды дисплея						
–	MSG	DISPLAY MESSAGE	–	–	–	–
Команды памяти файлов						
–	FREAD	READ DATA FILE	FWRIT	WRITE DATA FILE	–	–
Команды управления часами						
–	CADD	CALENDAR ADD	CSUB	CALENDAR SUBTRACT	SEC	HOURS TO SECONDS
	HMS	SECONDS TO HOURS	DATE	CLOCK ADJUSTMENT	–	–
Команды отладки программы						
–	TRSM	TRACE MEMORY SAMPLING	–	–	–	–
Команды диагностики отказов						
–	FAL	FAILURE ALARM	FALS	SEVERE FAILURE ALARM	FPD	FAILURE POINT DETECTION
Прочие команды						
–	STC	SET CARRY	CLC	CLEAR CARRY	EMBC	SELECT EM BANK
	WDT	EXTEND MAXIMUM CYCLE TIME	–	–	–	–
Команды программирования блоков						
Определение области блочной программы	BPRG	BLOCK PROGRAM BEGIN	BEND	BLOCK PROGRAM END	–	–
Выполнение от- ветвления IF	Адрес бита IF	CONDITIONAL BLOCK BRANCHING	Адрес бита IF NOT	CONDITIONAL BLOCK BRANCHING (NOT)	ELSE	CONDITIONAL BLOCK BRANCHING (ELSE)
	IEND	CONDITIONAL BLOCK BRANCHING END	–	–	–	–
WAIT	Адрес бита WAIT	ONE CYCLE AND WAIT	Адрес бита WAIT NOT	ONE CYCLE AND WAIT NOT	Условие ввода WAIT	ONE CYCLE AND WAIT
EXIT	Адрес бита EXIT	Условное окончание END	Адрес бита EXIT NOT	Условное продолжение END NOT	Условие ввода EXIT	Условное окончание END
Повторение	LOOP	LOOP BLOCK	Адрес бита LEND	LOOP BLOCK END	Адрес бита LEND NOT	LOOP BLOCK END NOT

Классификация	Мнемоника	Команда	Мнемоника	Команда	Мнемоника	Команда
Подкласс						
	Условие ввода	LOOP BLOCK	–	–	–	–
	LEND	END				
Запуск/остановка блока программы	BPPS	BLOCK PROGRAM PAUSE	BPRS	BLOCK PROGRAM RESTART	–	–
Таймер/счетчик	TIMW	TIMER WAIT	CNTW	COUNTER WAIT	TMHW	HIGH-SPEED TIMER WAIT
Команды обработки текстовых строк программы						
–	MOV\$	MOV STRING	+\$	CONCATENATE STRING	LEFT\$	GET STRING LEFT
	RIGHT\$	GET STRING RIGHT	MID\$	GET STRING MIDDLE	FIND\$	FIND IN STRING
	LEN\$	STRING LENGTH	RPLC\$	REPLACE IN STRING	DEL\$	DELETE STRING
	XCHG\$	EXCHANGE STRING	CLR\$	CLEAR STRING	INS\$	INSERT INTO STRING
	LD, AND, OR +=\$, <>\$, <\$, J\$, >\$, i\$	STRING COMPARISON	–	–	–	–
Команды управления задачами						
–	TKON	TASK ON	TKOF	TASK OFF	–	–

## 2-2 Функции команд

### 2-2-1 Команды последовательного ввода

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
LOAD	LD @LD %LD !LD !@LD !%LD		Обозначает логический старт и создает Условие выполнения ON/OFF, базируясь на противоположное состояние бита (ON/OFF), указанного в операнде.	Не требуется	85
LOAD NOT	LD NOT !LD NOT		Обозначает логический старт и создает Условие выполнения ON/OFF, базируясь на состояние бита (ON/OFF), указанного в операнде.	Не требуется	87
AND	AND @AND %AND !AND !@AND !%AND		Выполняет логическую операцию "И" с состоянием бита указанного операнда и текущим условием выполнения	Требуется	89
AND NOT	AND NOT !AND NOT		Изменяет состояние бита указанного операнда на противоположное и выполняет логическую операцию "И" с текущим условием выполнения	Требуется	90
OR	OR @OR %OR !OR !@OR !%OR		Выполняет логическую операцию "ИЛИ" с состоянием бита указанного операнда (ON/OFF) и текущим условием выполнения	Требуется	92
OR NOT	OR NOT !OR NOT		Изменяет состояние бита указанного операнда на противоположное и выполняет операцию "ИЛИ" с текущим условием выполнения	Требуется	93
AND LOAD	AND LD		Выполняет операцию "И" с логическими блоками	Требуется	95
OR LOAD	OR LD		Выполняет операцию "ИЛИ" с двумя логическими блоками	Требуется	97
NOT	NOT	520	Переводит условие выполнения в противоположное состояние	Требуется	102

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
CONDITION ON	UP	521	UP (521) переводит условие выполнения в состояние ON на время выполнения одного цикла, когда условие выполнения переходит из состояния OFF в состояние ON	Требуется	103
CONDITION OFF	DOWN	522	DOWN (522) переводит условие выполнения в состояние ON на время выполнения одного цикла, когда условие выполнения переходит из состояния ON в состояние OFF	Требуется	103
BIT TEST	LD TST	350	LD TST (350), AND TST (350), OR TST (350), используются в программе подобно командам LD, AND и OR; условие выполнения переводится в состояние ON когда заданный бит в заданном слове находится в состоянии ON. Условие выполнения переводится в состояние OFF, когда заданный бит в заданном слове находится в состоянии OFF	Не требуется	104
BIT TEST	LD TSTN	351	LD TSTN (351) AND TSTN (351) OR TSTN (351) используются в программе подобно командам LD NOT, AND NOT и OR NOT; условие выполнения переводится в состояние OFF когда заданный бит в заданном слове находится в состоянии ON. Условие выполнения переводится в состояние ON, когда заданный бит в заданном слове находится в состоянии OFF.	Не требуется	104
BIT TEST	AND TST	350	LD TSTN (351) AND TSTN (351) OR TSTN (351) используются в программе подобно командам LD NOT, AND NOT и OR NOT; условие выполнения переводится в состояние OFF когда заданный бит в заданном слове находится в состоянии ON. Условие выполнения переводится в состояние ON, когда заданный бит в заданном слове находится в состоянии OFF.	Требуется	104
BIT TEST	AND TSTN	351	LD TSTN (351), AND TSTN (351), OR TSTN (351) используются в программе подобно командам LD NOT, AND NOT и OR NOT; условие выполнения переводится в состояние OFF когда заданный бит в заданном слове находится в состоянии ON. Условие выполнения переводится в состояние ON, когда заданный бит в заданном слове находится в состоянии OFF.	Требуется	104
BIT TEST	OR TST	350	LD TST (350), AND TST (350), OR TST (350), используются в программе подобно командам LD, AND и OR; условие выполнения переводится в состояние ON когда заданный бит в заданном слове находится в состоянии ON. Условие выполнения переводится в состояние OFF, когда заданный бит в заданном слове находится в состоянии OFF.	Требуется	104
BIT TEST	OR TSTN	351	LD TSTN (351) AND TSTN (351) OR TSTN (351) используются в программе подобно командам LD NOT, AND NOT и OR NOT; условие выполнения переводится в состояние OFF когда заданный бит в заданном слове находится в состоянии ON. Условие выполнения переводится в состояние ON, когда заданный бит в заданном слове находится в состоянии OFF.	Требуется	104

## 2-2-2 Команды последовательного вывода

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
OUTPUT	OUT !OUT		Осуществляет вывод результата (состояния выполнения**) логической операции в заданный бит	Вывод Требуется	107
OUTPUT NOT	OUT NOT !OUT NOT		Переводит результат выполнения логической операции (состояния выполнения) в противоположное состояние и осуществляет его вывод в заданный бит	Вывод Требуется	108
KEEP	KEEP !KEEP	011	Функционирует подобно реле с самоудержанием	Вывод Требуется	109



Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
DIFFERENTIATE UP	DIFU !DIFU	013	Команда DIFU (013) переводит бит назначения в состояние ON на время выполнения одного цикла, когда условие выполнения переходит из состояния OFF в состояние ON (восходящий переход)	Вывод Требуется	113
DIFFERENTIATE DOWN	DIFD !DIFD	014	Команда DIFD (014) переводит бит назначения в состояние ON на время выполнения одного цикла, когда условие выполнения переходит из состояния ON в состояние OFF (ниспадающий переход)	Вывод Требуется	113
SET	SET @SET %SET !SET !@ SET !% SET		Команда SET переводит бит операнда в состояние ON, когда условие выполнения находится в состоянии ON	Вывод Требуется	115
RESET	RSET @RSET %RSET !RSET !@RSET !% RSET		Команда RESET переводит бит операнда в состояние ON, когда условие выполнения находится в состоянии ON	Вывод Требуется	115
MULTIPLE BIT SET	SETA @SETA	530	Команда SETA (530) переводит в состояние ON заданное количество последовательных битов	Вывод Требуется	117
MULTIPLE BIT RESET	RSTA @RSTA	531	Команда RSTA (531) переводит в состояние OFF заданное количество последовательных битов	Вывод Требуется	117

## 2-2-3 Команды последовательного управления

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
END	END	001	Обозначает окончание программы. Команда END (001) завершает выполнение программы в данном цикле. После команды END (001) не выполняется ни одна из записанных команд. Контроллер переходит к выполнению программы, имеющей следующий порядковый номер задачи. После выполнения задачи, имеющей самый большой порядковый номер в программе, команда END (001) обозначает завершение всей программы.	Вывод Не требуется	120
NO OPERATION	NOP	000	Данная команда не обладает функцией. (По команде NOP (000) не выполняется ни одно из действий)	Вывод Не требуется	121
INTERLOCK	IL	002	Осуществляет блокирование всех выводов, указанных между командами IL (002) и ILC (003), когда условием выполнения для команды IL (002) является OFF. Команды IL (002) и ILC (003) используются парами.	Вывод Требуется	121
INTERLOCK CLEAR	ILC	003	Осуществляет блокирование всех выводов, указанных между командами IL (002) и ILC (003), когда условием выполнения для команды IL (002) является OFF. Команды IL (002) и ILC (003) используются парами.	Вывод Не требуется	121
JUMP	JMP	004	Когда условием выполнения команды JMP (004) является OFF, выполнение программы переходит к первой команде JME (005) в программе с тем же порядковым номером перехода. Команды JMP (004) и JME (005) используются парами.	Вывод Требуется	125
JUMP END	JME	005	Обозначает завершение перехода, указанного командой JMP (004) или CJP (510)	Вывод Не требуется	125
CONDITIONAL JUMP	CJP	510	Выполнение команды CJP (510) по сути, противоположно выполнению команды JMP (004). Когда условием выполнения для CJP (510) является ON, программа переходит к выполнению первой команды JME (005) в программе с тем же порядковым номером перехода. Команды CJP (510) и JME (005) используются парами.	Вывод Требуется	128

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
CONDITIONAL JUMP	CJPN	511	Выполнение команды CJPN (511) практически идентично выполнению команды JMP (004). Когда условием выполнения для CJPN (004)** является OFF, выполнение программы переходит к первой команде JME (005) в программе с тем же порядковым номером перехода. Команды CJPN (511) и JME (005) используются парами.	Вывод Не требуется	128
MULTIPLE JUMP	JMP0	515	Когда условием выполнения команды JMP0 (515) является OFF, все команды от JMP0 (515) до JME0 (516) не выполняются, как при команде NOP (000). Команды JMP0 (515) и JME0 (516) используются парами. Ограничения на количество применяемых пар команд не устанавливаются.	Вывод Требуется	131
MULTIPLE JUMP END	JME0	516	Когда условием выполнения команды JMP0 (515) является OFF, все команды от JMP0 (515) до JME0 (516) не выполняются, как при команде NOP (000). Команды JMP0 (515) и JME0 (516) используются парами. Ограничения на количество применяемых пар команд не устанавливаются.	Вывод Требуется	131
FOR-NEXT LOOPS	FOR	512	Все команды, начиная от FOR (512) до NEXT (513) повторяются заданной количество раз. Команды FOR (512) и NEXT (513) используются парами.	Вывод Не требуется	133
BREAK LOOP	BREAK	514	Программируется в цикле FOR-NEXT для отмены выполнения цикла при заданном условии выполнения. Оставшиеся команды в цикле не выполняются, как при команде NOP (000).	Вывод Требуется	137
FOR-NEXT LOOPS	NEXT	513	Все команды, начиная от FOR (512) до NEXT (513) повторяются заданное количество раз. Команды FOR (512) и NEXT (513) используются парами.	Вывод Не требуется	133

## 2-2-4 Команды управления таймером и счетчиком

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
TIMER	TIM		Команда TIM приводит в действие таймер с отрицательным приращением величины (таймер с обратным отсчетом) в единицах, равных 0.1 сек. Диапазон задаваемых значений - от 0 до 999.9 сек.	Вывод Требуется	139
COUNTER	CNT		Команда CNT приводит в действие счетчик с отрицательным приращением счета (счетчик с обратным отсчетом). Диапазон задаваемых значений - от 0 до 9999.	Вывод Требуется	158
HIGH-SPEED TIMER	TIMH	015	Команда TIMH (015) приводит в действие высокоскоростной таймер с отрицательным приращением величины (таймер с обратным отсчетом) в единицах, равных 10 мсек. Диапазон задаваемых значений - от 0 до 99.99 сек.	Вывод Требуется	142
ONE-MS TIMER	TMHN	540	Команда TMHN (015) приводит в действие высокоскоростной таймер с отрицательным приращением величины (таймер с обратным отсчетом) в единицах, равных 1 мсек. Диапазон задаваемых значений - от 0 до 9.999 сек.	Вывод Требуется	145
ACCUMULATIVE TIMER	TTIM	087	Команда TTIM (087) приводит в действие таймер с приращением величины (таймер с прямым отсчетом) в единицах, равных 0.1 сек. Диапазон задаваемых значений - от 0 до 999.9 сек.	Вывод Требуется	147
LONG TIMER	TIML	542	Команда TIML (542) приводит в действие таймер с отрицательным приращением величины (таймер с обратным отсчетом) в единицах, равных 0.1 сек. Диапазон задаваемых значений - до 9999999.9 сек. (Примерно 115 дней.)	Вывод Требуется	151
MULTI-OUTPUT TIMER	MTIM	543	Команда MTIM (543) приводит в действие таймер с приращением величины (таймер с прямым отсчетом) с восемью независимыми заданными значениями и флагами завершения. Диапазон задаваемых значений - от 0 до 999.9 сек.	Вывод Требуется	153
REVERSIBLE COUNTER	CNTR	012	Команда CNTR (012) приводит в действие реверсивный счетчик	Вывод Требуется	160

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
RESET TIMER/ COUNTER	CNR @CNR	545	Осуществляет сброс таймеров или счетчиков в заданном диапазоне номеров таймеров или счетчиков. Устанавливает заданное значение (SV) в максимальное значение 9999.	Вывод Требуется	164

## 2-2-5 Команды сравнения

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
Symbol Comparison (символьные команды сравнения, без знака)					
	=	300	Символьные команды сравнения (без знака) осуществляют сравнение двух величин (констант и /или содержания заданных слов) в двоичном виде (длина данных - 16 битов) и создают состояние выполнения ON, когда результат сравнения верен. Существует три типа символьных команд сравнения: LD (LOAD), AND, OR.	LD: не требуется. AND, OR: требуется	175
	<>	305			
	<	310			
	<=	315			
	>	320			
	>=	325			
Symbol Comparison (символы сравнения, двойные слова без знака)					
	=	301	Символьные команды сравнения (двойные слова, без знака) осуществляют сравнение двух величин (констант и /или содержания заданных двойных слов) в двоичном виде (длина данных - 32 бита) и создают состояние выполнения ON, когда результат сравнения верен. Существует три символьных команд сравнения: LD (LOAD), AND, OR.	LD: не требуется. AND, OR: требуется	175
	<>	306			
	<	311			
	<=	316			
	>	321			
	>=	326			
Symbol Comparison (символы сравнения, со знаком)					
	=	302	Символьные команды сравнения (со знаком) осуществляют сравнение двух величин со знаками (констант и /или содержания заданных слов) в двоичном виде (длина данных - 32 бита) и создают состояние выполнения ON, когда результат сравнения верен. Существует три типа символьных команд сравнения: LD (LOAD), AND, OR.	LD: не требуется. AND, OR: требуется	175
	<>	307			
	<	312			
	<=	317			
	>	322			
	>=	327			
Symbol Comparison (символы сравнения, двойные слова без знака)					
	=	303	Символьные команды сравнения (двойные слова, со знаком) осуществляют сравнение двух величин (констант и /или содержания заданных двойных слов) в двоичном виде со знаком (длина данных - 32 бита, 8 шестн. цифр) и создают состояние выполнения ON, когда результат сравнения верен. Существует три типа символьных команд сравнения: LD (LOAD), AND, OR.	LD: не требуется. AND, OR: требуется	175
	<>	308			
	<	313			
	<=	318			
	>	323			
	>=	328			
COMPARE	CMR !CMR	020	Осуществляет сравнение двух двоичных значений без знака (констант или содержания заданных слов) и выводит результат в арифметические флаги вспомогательной области.	Вывод Требуется	180
DOUBLE COMPARE	CMPL	060	Осуществляет сравнение двух двойных двоичных значений без знака (констант или содержания заданных слов) и выводит результат в арифметические флаги вспомогательной области.	Вывод Требуется	182
SIGNED BINARY COMPARE	CPS !CPS	114	Осуществляет сравнение двух двоичных значений со знаком (констант или содержания заданных слов) и выводит результат в арифметические флаги вспомогательной области.	Вывод Требуется	185
DOUBLE SIGNED BINARY COMPARE	CPSL	115	Осуществляет сравнение двух двойных двоичных значений со знаком (констант или содержания заданных слов) и выводит результат в арифметические флаги вспомогательной области.	Вывод Требуется	187
TABLE COMPARE	TCMP @TCMP	085	Осуществляет сравнение исходных данных с содержанием 16-ти последовательных слов и переводит в состояние ON соответствующий бит результирующего слова, когда содержания слов равны.	Вывод Требуется	192
MULTIPLE COMPARE	MCMP @MCMP	019	Осуществляет сравнение 16-ти последовательных слов с другими 16-ю последовательных слов и переводит в состояние ON соответствующий бит результирующего слова, когда содержания слов не равны.	Вывод Требуется	190

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
BLOCK COMPARE	BCMP @BCMP	068	Осуществляет сравнение исходных данных с 16-ю диапазонами (определяемыми нижними и верхними пределами) и переводит в состояние ON соответствующий бит результирующего слова, когда исходные данные входят в указанный диапазон.	Вывод Требуется	195

## 2-2-6 Команды перемещения данных

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
MOVE	MOV @MOV !MOV !@MOV	021	Осуществляет передачу одного слова данных в заданное слово	Вывод Требуется	198
DOUBLE MOVE	MOVL @MOVL	498	Осуществляет передачу двух слов данных в заданные слова	Вывод Требуется	200
MOVE NOT	MVN @MVN	022	Осуществляет передачу дополнения одного слова данных в заданное слово	Вывод Требуется	199
DOUBLE MOVE NOT	MVNL @MVNL,	499	Осуществляет передачу дополнений двух слов данных в заданные слова	Вывод Требуется	202
MOVE BIT	MOVNB @MOVNB	082	Осуществляет передачу заданного бита	Вывод Требуется	203
MOVE DIGIT	MOVD @MOVD	083	Осуществляет передачу заданного однозначного числа или заданных однозначных чисел. (Каждое однозначное число состоит из 4-х битов.)	Вывод Требуется	205
MULTIPLE BIT TRANSFER	XFRB @XFRB	062	Осуществляет передачу заданного количества последовательных битов.	Вывод Требуется	208
BLOCK TRANSFER	XFEB @XFEB	070	Осуществляет передачу заданного количества последовательных слов	Вывод Требуется	210
BLOCK SET	BSET @BSET	071	Осуществляет копирование одного слова в диапазон последовательных слов	Вывод Требуется	212
DATA EXCHANGE	XCHG @XCHG	073	Осуществляет обмен содержания двух заданных слов	Вывод Требуется	214
DOUBLE DATA EXCHANGE	XCGL @XCGL	562	Осуществляет обмен содержания пары последовательных слов с другой парой последовательных слов	Вывод Требуется	215
SINGLE WORD DISTRIBUTE	DIST @DIST	080	Осуществляет передачу исходного слова в слово назначения (адресата), вычисляемого путем прибавления величины смещения к основному адресу	Вывод Требуется	217
DATA COLLECT	COLL @COLL	081	Осуществляет передачу исходного слова (вычисляемого путем прибавления величины смещения к основному адресу) в слово назначения (адресата)	Вывод Требуется	219
MOVE TO REGISTER	MOVR @MOVR	560	Устанавливает в заданном индексном регистре адрес памяти внутреннего ввода/вывода для заданного слова, бита, или флага завершения таймера/счетчика. Для установки адреса памяти внутреннего ввода/вывода для заданного текущего значения таймера или счетчика используйте команду MOVR(561)	Вывод Требуется	221
MOVE TIMER/COUNTER PV REGISTER	MOVRW @MOVRW	561	Устанавливает в заданном индексном регистре адрес памяти внутреннего ввода/вывода для заданного текущего значения таймера или счетчика. Для задания адреса памяти внутреннего ввода/вывода слова, бита, или флага завершения таймера/счетчика установки команду MOVRW (560)	Вывод Требуется	223

## 2-2-7 Команды сдвига данных

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
SHIFT REGISTER	SFT	010	Приводит в действие регистр сдвига	Вывод Требуется	225
REVERSIBLE SHIFT REGISTER	SFTR @SFTR	084	Создает регистр сдвига для смещения данных вправо или влево	Вывод Требуется	227
ASYNCHRONOUS SHIFT REGISTER	ASFT @ASFT	017	Осуществляет сдвиг всех ненулевых слов в заданном диапазоне слов	Вывод Требуется	229

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
WORD SHIFT	WSFT @WSFT	016	Осуществляет сдвиг всех данных между St и E в единицах, равных одному слову	Вывод Требуется	232
ARITHMETIC SHIFT LEFT	ASL @ASL	025	Осуществляет сдвиг содержания слова на один бит влево	Вывод Требуется	233
DOUBLE SHIFT LEFT	ASLL @ASLL	570	Осуществляет сдвиг содержания двух слов на один бит влево	Вывод Требуется	235
ARITHMETIC SHIFT RIGHT	ASR, @ASR	026	Осуществляет сдвиг содержания слова Wd на один бит вправо.	Вывод Требуется	237
DOUBLE SHIFT RIGHT	ASRL @ASRL	571	Осуществляет сдвиг содержания слов Wd и Wd + 1 на один бит вправо.	Вывод Требуется	238
ROTATE LEFT	ROL @ROL	027	Осуществляет сдвиг всех битов слова Wd на один бит влево, включая флаг переноса (CY).	Вывод Требуется	240
DOUBLE ROTATE LEFT	ROLL @ROLL	572	Осуществляет сдвиг всех битов слов Wd и Wd+1 на один бит влево, включая флаг переноса (CY).	Вывод Требуется	242
ROTATE LEFT WITHOUT CARRY	RLNC @RLNC	574	Осуществляет сдвиг всех битов слова Wd на один бит влево, исключая флаг переноса (CY).	Вывод Требуется	247
DOUBLE ROTATE LEFT WITHOUT CARRY	RLNL @RLNL	576	Осуществляет сдвиг всех битов слов Wd и Wd+1 на один бит влево, исключая флаг переноса (CY).	Вывод Требуется	248
ROTATE RIGHT	ROR @ROR	028	Осуществляет сдвиг всех битов слова Wd на один бит вправо, включая флаг переноса (CY).	Вывод Требуется	243
DOUBLE ROTATE LEFT	RORL @RORL	573	Осуществляет сдвиг всех битов слов Wd и Wd+1 на один бит вправо, включая флаг переноса (CY).	Вывод Требуется	245
ROTATE RIGHT WITHOUT CARRY	RRNC @RRNC	575	Осуществляет сдвиг всех битов слова Wd на один бит вправо, исключая флаг переноса (CY). Содержание крайнего правого бита слова Wd смещается в крайний левый бит к флагу переноса (CY).	Вывод Требуется	250
DOUBLE ROTATE RIGHT WITHOUT CARRY	RRNL @RRNL	577	Осуществляет сдвиг всех битов слов Wd и Wd+1 на один бит вправо, исключая флаг переноса (CY). Содержание крайнего правого бита слова Wd+1 смещается в крайний левый бит слова Wd и к флагу переноса (CY).	Вывод Требуется	251
ONE DIGIT SHIFT LEFT	SLD @SLD	074	Осуществляет сдвиг данных на одну цифру (4 бита) влево.	Вывод Требуется	253
ONE DIGIT SHIFT RIGHT	SRD @SRD	075	Осуществляет сдвиг данных на одну цифру (4 бита) вправо.	Вывод Требуется	255
SHIFT N-BIT DATA LEFT	NSFL @NSFL	578	Осуществляет сдвиг заданного количества битов влево.	Вывод Требуется	256
SHIFT N-BIT DATA RIGHT	NSFR @NSFR	579	Осуществляет сдвиг заданного количества битов вправо.	Вывод Требуется	258
SHIFT N-BITS LEFT	NASL @NASL	580	Осуществляет сдвиг заданных 16 битов слова данных влево, на заданное количество битов.	Вывод Требуется	260
DOUBLE SHIFT N-BITS LEFT	NSLL @NSLL	582	Осуществляет сдвиг заданных 32 битов слова данных влево, на заданное количество битов.	Вывод Требуется	263
SHIFT N-BITS RIGHT	NASR @NASR	581	Осуществляет сдвиг заданных 16 битов слова данных вправо, на заданное количество битов.	Вывод Требуется	266
DOUBLE SHIFT N-BITS RIGHT	NSRL @NSRL	583	Осуществляет сдвиг заданных 32 битов слова данных вправо, на заданное количество битов.	Вывод Требуется	269

## 2-2-8 Команды увеличения, уменьшения

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
INCREMENT BINARY	++ @++	590	Осуществляет увеличение заданного слова, состоящего из четырехзначного шестнадцатеричного числа, на единицу	Вывод Требуется	272
DOUBLE INCREMENT BINARY	++L @++L	591	Осуществляет увеличение заданного слова, состоящего из восьмизначного шестнадцатеричного числа, на единицу	Вывод Требуется	273
DECREMENT BINARY	-- @--	592	Осуществляет уменьшение заданного слова, состоящего из четырехзначного шестнадцатеричного числа, на единицу	Вывод Требуется	275

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
DOUBLE DECREMENT BINARY	--L @--L	593	Осуществляет уменьшение заданного слова, состоящего из восьмизначного шестнадцатеричного числа, на единицу	Вывод Требуется	277
INCREMENT BCD	++B @++B	594	Осуществляет увеличение заданного слова, состоящего из четырехзначного двоично-десятичного числа, на единицу	Вывод Требуется	279
DOUBLE INCREMENT BCD	++BL @++BL	595	Осуществляет увеличение заданного слова, состоящего из восьмизначного двоично-десятичного числа, на единицу	Вывод Требуется	281
DECREMENT BCD	--B @--B	596	Осуществляет уменьшение заданного слова, состоящего из четырехзначного двоично-десятичного числа, на единицу	Вывод Требуется	283
DOUBLE DECREMENT BCD	--BL @--BL	597	Осуществляет уменьшение заданного слова, состоящего из восьмизначного двоично-десятичного числа, на единицу	Вывод Требуется	285

## 2-2-9 Символьные математические команды

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
SIGNED BINARY ADD WITHOUT CARRY	+ @+	400	Осуществляет сложение четырехзначных шестнадцатеричных данных (одно слово) и /или констант	Вывод Требуется	288
DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+L @+L	401	Осуществляет сложение восьмизначных шестнадцатеричных данных (двойное слово) и/или констант.	Вывод Требуется	290
SIGNED BINARY ADD WITH CARRY	+C @+C	402	Осуществляет сложение четырехзначных шестнадцатеричных данных (одно слово) и /или констант с флагом переноса (CY)	Вывод Требуется	292
DOUBLE SIGNED BINARY ADD WITH CARRY	+CL @+CL	403	Осуществляет сложение восьмизначных шестнадцатеричных данных (двойное слово) и/или констант, с флагом переноса (CY)	Вывод Требуется	294
BCD ADD WITHOUT CARRY	+B @+B	404	Осуществляет сложение четырехзначных двоично-десятичных данных (одно слово) и /или констант	Вывод Требуется	295
DOUBLE BCD ADD WITHOUT CARRY	+BL @+BL	405	Осуществляет сложение восьмизначных шестнадцатеричных данных (двойное слово) и/или констант	Вывод Требуется	297
BCD ADD WITH CARRY	+BC @+BC	406	Осуществляет сложение четырехзначных двоично-десятичных данных (одно слово) и /или констант, с флагом переноса (CY)	Вывод Требуется	299
DOUBLE BCD ADD WITH CARRY	+BCL @+BCL	407	Осуществляет сложение восьмизначных шестнадцатеричных данных (двойное слово) и/или констант, с флагом переноса (CY)	Вывод Требуется	300
SIGNED BINARY SUBSTRACT WITHOUT CARRY	- @-	410	Осуществляет вычитание четырехзначных шестнадцатеричных данных (одно слово) и /или констант	Вывод Требуется	302
DOUBLE SIGNED BINARY SUBSTRACT WITHOUT CARRY	-L @-L	411	Осуществляет вычитание восьмизначных шестнадцатеричных данных (двойное слово) и/или констант	Вывод Требуется	304
SIGNED BINARY SUBSTRACT WITH CARRY	-C @-C	412	Осуществляет вычитание четырехзначных шестнадцатеричных данных (одно слово) и /или констант, с флагом переноса (CY)	Вывод Требуется	307
DOUBLE SIGNED BINARY SUBSTRACT WITH CARRY	-CL @-CL	413	Осуществляет вычитание восьмизначных шестнадцатеричных данных (двойное слово) и/или констант, с флагом переноса (CY)	Вывод Требуется	309
BCD SUBSTRACT WITHOUT CARRY	-B @-B	414	Осуществляет вычитание четырехзначных двоично-десятичных данных (одно слово) и /или констант	Вывод Требуется	312



Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
DOUBLE BCD SUBSTRACT WITHOUT CARRY	-BL @-BL	415	Осуществляет вычитание восьмизначных двоично-десятичных данных (двойное слово) и/или констант	Вывод Требуется	313
BCD SUBSTRACT WITH CARRY	-BC @-BC	416	Осуществляет вычитание четырехзначных двоично-десятичных данных (одно слово) и /или констант с флагом переноса (CY)	Вывод Требуется	316
DOUBLE BCD SUBSTRACT WITH CARRY	-BCL @-BCL	417	Осуществляет вычитание восьмизначных двоично-десятичных данных (двойное слово) и /или констант, с флагом переноса (CY)	Вывод Требуется	318
SIGNED BINARY MULTIPLY	* @*	420	Осуществляет умножение четырехзначных шестнадцатеричных данных и /или констант	Вывод Требуется	320
DOUBLE SIGNED BINARY MULTIPLY	*L @*L	421	Осуществляет умножение восьмизначных шестнадцатеричных данных и /или констант	Вывод Требуется	321
UNSIGNED BINARY MULTIPLY	*U @*U	422	Осуществляет умножение четырехзначных шестнадцатеричных данных и /или констант	Вывод Требуется	323
DOUBLE SIGNED BINARY MULTIPLY	*UL @*UL	423	Осуществляет умножение восьмизначных шестнадцатеричных данных и /или констант	Вывод Требуется	324
BCD MULTIPLY	*B @*B	424	Осуществляет умножение четырехзначных (одно слово) двоично-десятичных данных и /или констант	Вывод Требуется	326
DOUBLE BCD MULTIPLY	*BL @*BL	425	Осуществляет умножение восьмизначных (двойное слово) двоично-десятичных данных и /или констант	Вывод Требуется	327
SIGNED BINARY DIVIDE	/ @/	430	Осуществляет деление четырехзначных (одно слово) шестнадцатеричных данных и /или констант	Вывод Требуется	329
DOUBLE SIGNED BINARY DIVIDE	/L @/L	431	Осуществляет деление восьмизначных (двойное слово) шестнадцатеричных данных и /или констант	Вывод Требуется	330
UNSIGNED BINARY DIVIDE	/U @/U	432	Осуществляет деление четырехзначных (одно слово) шестнадцатеричных данных без знака и /или констант	Вывод Требуется	332
DOUBLE UNSIGNED BINARY DIVIDE	/UL @/UL	433	Осуществляет деление восьмизначных (двойное слово) шестнадцатеричных данных и /или констант	Вывод Требуется	334
BCD DIVIDE	/B @/B	434	Осуществляет деление четырехзначных (одно слово) двоично-десятичных данных и /или констант	Вывод Требуется	335
DOUBLE BCD DIVIDE	/BL @/BL	435	Осуществляет деление восьмизначных (двойное слово) двоично-десятичных данных и /или констант	Вывод Требуется	337

## 2-2-10 Команды преобразования

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
BCD-TO- BINARY	BIN @BIN	023	Осуществляет преобразование двоично-десятичных данных в двоичные данные	Вывод Требуется	339
DOUBLE BCD-TO- DOUBLE BINARY	BINL @BINL	058	Осуществляет преобразование восьмизначных двоично-десятичных данных в восьмизначные шестнадцатеричные данные (32-х битовые двоичные данные)	Вывод Требуется	340
BINARY-TO- BCD	BCD @BCD	024	Осуществляет преобразование одного слова двоичных данных в одно слово двоично-десятичных данных	Вывод Требуется	342
DOUBLE BINARY -TO-DOUBLE BCD	BCDL @BCDL	059	Осуществляет преобразование восьмизначных шестнадцатеричных данных (32-х битовые двоичные данные) в восьмизначные двоично-десятичные данные	Вывод Требуется	343
2's COMPLEMENT	NEG @NEG	160	Осуществляет вычисление дополнения по базе 2 от одного слова шестнадцатеричных данных	Вывод Требуется	345
DOUBLE 2's COMPLEMENT	NEGL @NEGL	161	Осуществляет вычисление дополнения по базе 2 от двух слов шестнадцатеричных данных	Вывод Требуется	347
16 BIT TO 32 BIT SIGNED BINARY	SIGN @SIGN	600	Осуществляет преобразование 16-ти битовых двоичных со знаком данных в эквивалентные 32-х битовые двоичные данные	Вывод Требуется	349



Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
DATA DECODER	MLPX @MLPX	076	Осуществляет чтение числового значения в заданной цифре (или бите) исходного слова, переводит соответствующий бит результирующего слова (или диапазона из 16-ти слов) в состояние ON, и переводит все остальные биты результирующего слова (или диапазона из 16-ти слов) в состояние OFF.	Вывод Требуется	350
DATA ENCODER	DMPX @DMPX	077	Находит расположение первого или последнего бита в состоянии ON в исходном слове (или в диапазоне из 16-ти слов) и записывает это значение в заданную цифру (или байт) результирующего слова.	Вывод Требуется	355
ASCII CONVERT	ASC @ASC	086	Осуществляет преобразование 4-х битовых шестнадцатеричных данных в исходном слове в 8-ми битовые эквиваленты в ASCII коде.	Вывод Требуется	359
ASCII TO HEX	HEX @HEX	162	Осуществляет преобразование до 4-х байтов данных исходного слова в ASCII коде в их шестнадцатеричные эквиваленты и записывает эти цифры в указанные слова назначения.	Вывод Требуется	363
COLUMN TO LINE	LINE @LINE	063	Осуществляет преобразование колонки битов из диапазона слов размахом 16 слов (один и тот же номер бита в 16-ти последовательных словах) в 16 битов слова назначения.	Вывод Требуется	367
LINE TO COLUMN	COLM @COLM	064	Осуществляет преобразование 16-ти битов исходного слова в колонку битов в диапазоне слов назначения размахом 16 слов (один и тот же номер бита в 16-ти последовательных словах).	Вывод Требуется	369
SIGNED BCD TO BINARY	BINS @BINS	470	Осуществляет преобразование одного слова двоично-десятичных данных со знаком в одно слово двоичных данных со знаком.	Вывод Требуется	371
DOUBLE SIGNED BCD TO BINARY	BISL @BISL	472	Осуществляет преобразование двойных двоично-десятичных данных со знаком в двойные двоичные данные со знаком.	Вывод Требуется	374
SIGNED BINARY TO BCD	BCDS @BCDS	471	Осуществляет преобразование одного слова двоичных данных со знаком в одно слово двоично-десятичных данных со знаком.	Вывод Требуется	377
DOUBLE SIGNED BINARY TO BCD	BDSL @BDSL	473	Осуществляет преобразование двойных двоичных данных со знаком в двойные двоично-десятичные данные со знаком.	Вывод Требуется	379

## 2-2-11 Логические команды

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
LOGICAL AND	ANDW @ANDW	034	Выполняет логическую операцию "И" над соответствующими битами простых слов, данных в виде слов и/или константами.	Вывод Требуется	384
DOUBLE LOGICAL AND	ANDL @ANDL	610	Выполняет логическую операцию "И" над соответствующими битами двойных слов, данных в виде слов и/или константами.	Вывод Требуется	385
LOGICAL OR	ORW @ORW	035	Выполняет логическую операцию "ИЛИ" над соответствующими битами простых слов, данных в виде слов и/или константами.	Вывод Требуется	387
DOUBLE LOGICAL OR	ORWL @ORWL	611	Выполняет логическую операцию "ИЛИ" над соответствующими битами двойных слов, данных в виде слов и/или константами.	Вывод Требуется	388
EXCLUSIVE OR	XORW @XORW	036	Выполняет логическую операцию "исключающее ИЛИ" над соответствующими битами простых слов, данных в виде слов и/или константами.	Вывод Требуется	390
DOUBLE EXCLUSIVE OR	XORL @XORL	612	Выполняет логическую операцию "исключающее ИЛИ" над соответствующими битами двойных слов или данных в виде слов и/или константами.	Вывод Требуется	391
EXCLUSIVE NOR	XNRW @XNRW	037	Выполняет логическую операцию "исключающее НЕ-ИЛИ" над соответствующими битами простых слов или данных в виде слов и/или константами.	Вывод Требуется	393
DOUBLE EXCLUSIVE NOR	XNRL @XNRL	613	Выполняет логическую операцию "исключающее НЕ-ИЛИ" над соответствующими битами двойных слов или данных в виде слов и/или константами.	Вывод Требуется	395

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
COMPLEMENT	COM @COM	029	Осуществляет переключение всех битов из состояния ON в состояние OFF, и всех битов из состояния OFF в состояние ON в слове Wd.	Вывод Требуется	397
DOUBLE COMPLEMENT	COML @COML	614	Осуществляет переключение всех битов из состояния ON в состояние OFF, и всех битов из состояния OFF в состояние ON в словах Wd и Wd+1.	Вывод Требуется	398

## 2-2-12 Специальные математические команды

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
BINARY ROOT	ROTB @ROTB	620	Осуществляет вычисление квадратного корня из 32-х битового содержания заданных слов и вывод целого числа результата в заданное результирующее слово.	Вывод Требуется	400
BCD SQUARE ROOT	ROOT @ROOT	072	Осуществляет вычисление квадратного корня из восьмизначного двоично-десятичного числа и вывод целого числа результата в заданное результирующее слово.	Вывод Требуется	401
ARITHMETIC PROCESS	APR @APR	069	Осуществляет вычисление синуса, косинуса, или линейную экстраполяцию исходных данных. Функция линейной экстраполяции позволяет представить любую зависимость Y от X в виде отрезков прямых линий.	Вывод Требуется	405
FLOATING POINT DIVIDE	FDIV @FDIV	079	Осуществляет деление одного семизначного числа с плавающей запятой на другое. Числа с плавающей запятой выражены в виде: мантисса - 7 знаков, показатель - 1 знак.	Вывод Требуется	410
BIT COUNTER	BCNT @BCNT	067	Осуществляет подсчет общего количества битов, находящихся в состоянии ON в заданном слове (словах)	Вывод Требуется	414

## 2-2-13 Команды математических операций с плавающей запятой

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
FLOATING TO 16-BIT	FIX @FIX	450	Осуществляет преобразование 32-х битового числа с плавающей запятой в 16-ти битовые двоичные данные со знаком и записывает результат в заданное результирующее слово.	Вывод/ Требуется	421
FLOATING TO 32-BIT	FIXL @FIXL	451	Осуществляет преобразование 32-х битового числа с плавающей запятой в 32-х битовые двоичные данные со знаком и записывает результат в заданные результирующие слова.	Вывод/ Требуется	422
16-BIT TO FLOATING	FLT @FLT	452	Осуществляет преобразование 16-ти битового двоичного числа со знаком в 32-х битовые данные с плавающей запятой и записывает результат в заданные результирующие слова.	Вывод/ Требуется	424
32-BIT TO FLOATING	FLTL @FLTL	453	Осуществляет преобразование 32-х битового двоичного числа со знаком в 32-х битовые данные с плавающей запятой и записывает результат в заданные результирующие слова.	Вывод/ Требуется	425
FLOATING- POINT ADD	+F @+F	454	Осуществляет сложение двух 32-х битовых чисел с плавающей запятой и записывает результат в заданные результирующие слова.	Вывод/ Требуется	427
FLOATING- POINT SUBSTRACT	-F @-F	455	Осуществляет вычитание одного 32-х битового числа с плавающей запятой из другого и записывает результат в заданные результирующие слова.	Вывод/ Требуется	428
FLOATING- POINT DIVIDE	/F @/F	457	Осуществляет деление одного 32-х битового числа с плавающей запятой на другое и записывает результат в заданные результирующие слова.	Вывод/ Требуется	432
FLOATING- POINT MULTIPLY	*F @*F	456	Осуществляет умножение двух 32-х битовых чисел с плавающей запятой и записывает результат в заданные результирующие слова.	Вывод/ Требуется	430
DEGREES TO RADIANS	RAD @RAD	458	Осуществляет преобразование 32-х битового числа с плавающей запятой знаком из градусов в радианы и записывает результат в заданные результирующие слова.	Вывод/ Требуется	434

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
RADIANS TO DEGREES	DEG @DEG	459	Осуществляет преобразование 32-х битового числа с плавающей запятой из радианов в градусы и записывает результат в заданные результирующие слова.	Вывод/ Требуется	436
SINE	SIN @SIN	460	Осуществляет вычисление синуса 32-х битового числа с плавающей запятой (в радианах) и записывает результат в заданные результирующие слова.	Вывод/ Требуется	437
COSINE	COS @COS	461	Осуществляет вычисление косинуса 32-х битового числа с плавающей запятой (в радианах) и запись результата в заданные результирующие слова.	Вывод/ Требуется	439
TANGENT	TAN @TAN	462	Осуществляет вычисление тангенса 32-х битового числа с плавающей запятой (в радианах) и записывает результат в заданные результирующие слова.	Вывод/ Требуется	440
ARC SINE	ASIN @ASIN	463	Осуществляет вычисление арксинуса 32-х битового числа с плавающей запятой (в радианах) и записывает результат в заданные результирующие слова. (Функция вычисления арксинуса является обратной по отношению к вычислению синуса, она вычисляет угол, синус которого равен заданному значению от -1 до 1.)	Вывод/ Требуется	442
ARC COSINE	ACOS @ACOS	464	Осуществляет вычисление арккосинуса 32-х битового числа с плавающей запятой (в радианах) и записывает результат в заданные результирующие слова. (Функция вычисления арккосинуса является обратной по отношению к вычислению косинуса, она вычисляет угол, косинус которого равен заданному значению от -1 до 1.)	Вывод/ Требуется	444
ARC TANGENT	ATAN @ATAN	465	Осуществляет вычисление тангенса 32-х битового числа с плавающей запятой (в радианах) и записывает результат в заданные результирующие слова. (Функция вычисления арктангенса является обратной по отношению к вычислению тангенса, она вычисляет угол, тангенс которого равен заданному значению.)	Вывод/ Требуется	446
SQUARE ROOT	SQRT @SQRT	466	Осуществляет вычисление квадратного корня из 32-х битового числа с плавающей запятой и записывает результат в заданные результирующие слова.	Вывод/ Требуется	447
EXPONENT	EXP @EXP	467	Осуществляет вычисление натуральной экспоненты (по основанию e) 32-х битового числа с плавающей запятой и записывает результат в заданные результирующие слова.	Вывод/ Требуется	449
LOGARITHM	LOG @LOG	468	Осуществляет вычисление натурального логарифма (по основанию e) 32-х битового числа с плавающей запятой и записывает результат в заданные результирующие слова.	Вывод/ Требуется	451
EXPONENTIAL POWER	PWR @PWR	840	Осуществляет возведение 32-х битового числа с плавающей запятой в степень другого 32-х битового числа с плавающей запятой и записывает результат в заданные результирующие слова.	Вывод/ Требуется	453

## 2-2-14 Команды обработки табличных данных

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
SET STACK	SSET @SSET	630	Определяет стек заданной длины и начинающийся заданным словом. Инициализирует слова в области данных, присваивая им нулевые значения.	Вывод/ Требуется	458
PUSH ONTO STACK	PUSH @PUSH	632	Осуществляет запись одного слова в заданный стек.	Вывод/ Требуется	460
FIRST IN FIRST OUT	FIFO @FIFO	633	Осуществляет чтение первого слова данных, записанного в заданный стек (Наиболее устаревшие данные стека).	Вывод/ Требуется	463
LAST IN FIRST OUT	LIFO @LIFO	634	Осуществляет чтение последнего слова данных, записанного в заданный стек (Наиболее свежие данные стека).	Вывод/ Требуется	465
DIMENSION RECORD TABLE	DIM @DIM	631	Определяет таблицу записей путем задания длины каждой из записей и количества записей. Допускается задание до 16 таблиц записей.	Вывод/ Требуется	468

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
SET RECORD LOCATION	SETR @SETR	635	Осуществляет запись расположения заданной записи (адрес памяти внутреннего ввода/вывода начала записи) в заданный индексный регистр.	Вывод/ Требуется	470
GET RECORD NUMBER	GETR @GETR	636	Возвращает номер записи в таблице.	Вывод/ Требуется	472
DATA SEARCH	SRCH @SRCH	181	Осуществляет поиск слова в диапазоне слов.	Вывод/ Требуется	474
SWAP BYTES	SWAP @SWAP	637	Осуществляет обмен между крайним левым и крайним правым битами всех слов в заданном диапазоне.	Вывод/ Требуется	477
FIND MAXIMUM	MAX @MAX	182	Осуществляет поиск максимального значения в диапазоне слов.	Вывод/ Требуется	479
FIND MINIMUM	MIN @MIN	183	Осуществляет поиск минимального значения в диапазоне слов.	Вывод/ Требуется	482
SUM	SUM @SUM	184	Осуществляет сложение байтов или слов в диапазоне слов и вывод результата в два слова.	Вывод/ Требуется	485
FRAME CHECKSUM	FCS @FCS	180	Вычисляет значение контрольной суммы кадра в заданном диапазоне и записывает результат в ASCII коде.	Вывод/ Требуется	488

## 2-2-15 Команды управления данными

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
PID CONTROL	PID	190	Выполняет PID регулирование согласно заданным параметрам.	Вывод/ Требуется	492
LIMIT CONTROL	LMT @LMT	680	Осуществляет управление выходными данными согласно тому, входят ли входные данные в заданные пределы: верхний и нижний.	Вывод/ Требуется	502
DEAD BAND CONTROL	BAND @BAND	681	Осуществляет управление выходными данными согласно тому, входят ли входные данные в пределы диапазона нечувствительности.	Вывод/ Требуется	504
DEAD ZONE CONTROL	ZONE @ZONE	682	Добавляет смещение к входным данным и выводит результат.	Вывод/ Требуется	506
SCALING	SCL @SCL	194	Осуществляет преобразование двоичных данных без знака в двоично-десятичные данные без знака согласно заданной линейной зависимости.	Вывод/ Требуется	509
SCALING 2	SCL2 @SCL2	486	Осуществляет преобразование двоичных данных со знаком в двоично-десятичные данные со знаком согласно заданной линейной зависимости. В заданную линейную зависимость можно вводить смещение.	Вывод/ Требуется	513
SCALING 3	SCL3 @SCL3	487	Осуществляет преобразование двоично-десятичных данных со знаком в двоичные данные со знаком согласно заданной линейной зависимости. В заданную линейную зависимость можно вводить смещение.	Вывод/ Требуется	516
AVERAGE	AVG	195	Осуществляет вычисление среднего значения входного слова в заданном количестве циклов.	Вывод/ Требуется	520

## 2-2-16 Команды подпрограмм

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
SUBROUTINE CALL	SBS @SBS	091	Осуществляет вызов подпрограммы с заданным номером и выполняет эту программу.	Вывод/ Требуется	524
SUBROUTINE ENTRY	SBN	092	Обозначает начало подпрограммы с заданным номером.	Вывод/ Не требуется	533
SUBROUTINE RETURN	RET	093	Обозначает конец подпрограммы.	Вывод/ Не требуется	535
MACRO	MCRO @MCRO	099	Осуществляет вызов подпрограммы с заданным номером и выполняет эту программу, используя параметры ввода в S...S+3. и параметры вывода в D...D+3.	Вывод/ Требуется	529

## 2-2-17 Команды управления прерываниями

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
SET INTERRUPT MASK	MSKS @MSKS	690	Устанавливает процедуру прерывания ввода/вывода или прерывание по расписанию. Как задачи прерывания ввода/вывода, так и задачи прерывания по расписанию при первом включении Программируемого контроллера маскированы (блокированы). Команда MSKS(690) может использоваться для блокирования и разблокирования прерываний ввода/вывода, а также для установления интервалов для прерываний по расписанию.	Вывод/ Требуется	537
READ INTERRUPT MASK	MSKR @MSKR	692	Осуществляет чтение установок текущей процедуры прерывания, заданной командой MSKS(690).	Вывод/ Требуется	540
CLEAR INTERRUPT	CLI @CLI	691	Осуществляет очистку (сброс) или сохранение записанных вводов прерывания ввода/вывода или устанавливает время первого из прерываний по расписанию.	Вывод/ Требуется	542
DISABLE INTERRUPTS	DI @DI	693	Осуществляет блокировку выполнения всех задач прерывания, за исключением прерывания при отключении питания.	Вывод/ Требуется	545
ENABLE INTERRUPTS	EI @EI	694	Разрешает выполнение всех задач прерывания, блокированных командой DI(693).	Вывод/ Не требуется	546

## 2-2-18 Шаговые команды

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
STEP DEFINE	STEP	008	Команда STEP(008) функционирует двумя следующими способами, в зависимости от ее положения и от того, задан ли управляющий бит. Начинает выполнение заданного шага. Завершает область программирования шага (т.е. завершает выполнение шага).	Вывод/ Требуется	553
STEP START	SNXT	009	Команда SNXT(009) используется следующими тремя способами: Для запуска процедуры программирования шага. Для перехода к следующему управляющему биту. Завершает выполнение программирования шага	Вывод/ Требуется	553

## 2-2-19 Команды Базовых модулей ввода/вывода

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
I/O REFRESH	IORF @IORF	097	Осуществляет регенерацию заданных слов ввода/вывода.	Вывод/ Требуется	569
7-SEGMENT DECODER	SDEC @SDEC	078	Осуществляет преобразование содержания цифры адресата в шестнадцатеричном коде в восьмибитовый код семисегментного дисплея и размещает его в верхние или нижние 8 битов заданных слов назначения.	Вывод/ Требуется	571
INTELLIGENT I/O READ	IORD @IORD	222	Осуществляет чтение содержания области памяти Модуля ввода/вывода.	Вывод/ Требуется	574
INTELLIGENT I/O WRITE	IOWR @IOWR	223	Осуществляет вывод содержания области памяти ввода/вывода Модуля центрального процессора в Специальный модуль.	Вывод/ Требуется	577

## 2-2-20 Команды последовательного коммуникационного обмена

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
PROTOCOL MACRO	PMCR @PMCR	260	Осуществляет вызов и выполнение коммуникационной последовательности, зарегистрированной в Плате последовательного коммуникационного обмена или в Модуле последовательного коммуникационного обмена.	Вывод/ Требуется	582

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
TRANSMIT	TXD @TXD	236	Осуществляет вывод заданного количества байтов данных через порт RS-232C, встроенный в Модуль центрального процессора.	Вывод/ Требуется	589
RECEIVE	RXD @RXD	235	Осуществляет чтение заданного количества байтов данных через порт RS-232C, встроенный в Модуль центрального процессора.	Вывод/ Требуется	593
CHANGE SERIAL PORT SETUP	STUP @STUP	237	Осуществляет изменение коммуникационных параметров последовательного порта Модуля центрального процессора, Модуля последовательного коммуникационного обмена (Модуля шины центрального процессора) или Платы последовательного коммуникационного обмена. Таким образом, команда STUP (237) разрешает изменять протокол обмена в процессе работы Программируемого контроллера.	Вывод/ Требуется	598

## 2-2-21 Сетевые команды

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
NETWORK SEND	SEND @SEND	090	Осуществляет передачу данных в узел сети.	Вывод/ Требуется	606
NETWORK RECEIVE	RECV @RECV	098	Запрашивает данные, подлежащие передаче из узла сети, и принимает эти данные.	Вывод/ Требуется	611
DELIVER COMMAND	CMND @CMND	490	Осуществляет передачу команд FINS и принимает ответы.	Вывод/ Требуется	614

## 2-2-22 Команды памяти файлов

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
READ DATA FILE	FREAD @FREAD	700	Осуществляет чтение заданных данных или заданного количества данных из заданного файла в памяти файлов и запись в указанную область данных Модуля центрального процессора.	Вывод/ Требуется	621
WRITE DATA FILE	FWRIT @FWRIT	701	Осуществляет перезапись или добавление данных в заданном файле памяти файлов. Указанные данные берутся из области данных Модуля центрального процессора. Если указанный файл не существует, создается новый файл с этим именем.	Вывод/ Требуется	627

## 2-2-23 Команды дисплея

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
DISPLAY MESSAGE	MSG @MSG	046	Осуществляет чтение указанных шестнадцати слов в расширенном коде ASCII и вывод сообщения на дисплей Периферийного устройства, например Пульта программирования.	Вывод/ Требуется	635

## 2-2-24 Команды управления часами

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
CALENDAR ADD	CADD @CADD	730	Осуществляет прибавление времени к данным календаря в заданных словах.	Вывод/ Требуется	638
CALENDAR SUBSTRACT	CSUB @CSUB	731	Осуществляет вычитание времени из данных календаря в заданных словах.	Вывод/ Требуется	641
HOURS TO SECONDS	SEC @SEC	065	Осуществляет преобразование времени в формате "часы/минуты/секунды" в эквивалентное время, выраженное только в секундах.	Вывод/ Требуется	638
SECONDS TO HOURS	HMS @HMS	066	Осуществляет преобразование времени, выраженного только в секундах, в эквивалентное время в формате "часы/минуты/секунды".	Вывод/ Требуется	646
CLOCK AJUSTMENT	DATE @DATE	735	Осуществляет изменение установок внутренних часов на установки, заданные в исходных словах	Вывод/ Требуется	648



## 2-2-25 Команды отладки программы

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
TRACE MEMORY SAMPLING	TRSM	045	При выполнении команды TRSM (045), состояние предварительно выбранных битов или слов запоминается в памяти отслеживания. Команда TRSM (045) может использоваться в любом месте программы и сколько угодно раз.	Вывод/ Требуется	652

## 2-2-26 Команды диагностики отказов

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
FAILURE ALARM	FAL @FAL	006	Генерирует или сбрасывает определяемые пользователем допустимые ошибки. При возникновении допустимых ошибок работа Программируемого контроллера продолжается.	Вывод/ Требуется	656
SEVERE FAILURE ALARM	FALS @FALS	007	Генерирует определяемые пользователем критические ошибки. При возникновении критических ошибок работа Программируемого контроллера прекращается.	Вывод/ Требуется	659
FAILURE POINT DETECTION	FPD	269	Осуществляет диагностику отказа в блоке команд посредством мониторинга времени между выполнением команды FPD (269) и выводом сигнала диагностики и нахождением входа, препятствующего переводу вывода в состояние ON.	Вывод/ Требуется	662

## 2-2-27 Прочие команды

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
SET CARRY	STC @STC	040	Осуществляет установку флага переноса (CY).	Вывод/ Требуется	671
CLEAR CARRY	CLC @CLC	041	Осуществляет перевод флага переноса (CY) в состояние OFF.	Вывод/ Требуется	671
SELECT EM BANK	EMBC @EMBC	281	Осуществляет изменение текущего банка расширенной памяти (EM).	Вывод/ Требуется	672
EXTEND MAXIMUM CYCLE TIME	WDT @WDT	094	Осуществляет увеличение максимальной длительности цикла, однако только для цикла, в котором данная команда выполняется.	Вывод/ Требуется	674

## 2-2-28 Команды программирования блоков

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
BLOCK PROGRAM BEGIN	BPRG	096	Определяет область программирования блоков. Каждой из используемых команд BPRG(096) должна соответствовать команда BEND(801).	Вывод/ Требуется	681
BLOCK PROGRAM END	BEND	801	Определяет область программирования блоков. Каждой из применяемых команд BPRG(096) должна соответствовать команда BEND(801).	Вывод/ Требуется	681
BLOCK PROGRAM PAUSE	BPPS	811	Осуществляет временную остановку выполнения и последующий запуск заданного блока программы из другого блока программы.	Блок программы/ Требуется	684
BLOCK PROGRAM RESTART	BPRS	812	Осуществляет временную остановку выполнения и последующий запуск заданного блока программы из другого блока программы.	Блок программы/ Требуется	684
CONDITIONAL BLOCK EXIT	EXIT	806	Команда EXIT(806) без операнда-бита осуществляет выход из программы, если условие выполнения находится в состоянии ON.	Блок программы/ Требуется	691
CONDITIONAL BLOCK EXIT	EXIT	806	Команда EXIT(806) без операнда осуществляет выход из программы, если условие выполнения находится в состоянии ON.	Блок программы/ Требуется	691
CONDITIONAL BLOCK EXIT(NOT)	EXIT NOT	806	Команда EXIT(806) без операнда осуществляет выход из программы, если условие выполнения находится в состоянии ON.	Блок программы/ Требуется	691



Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
CONDITIONAL BLOCK BRANCHING	IF	802	Если условие выполнения находится в состоянии ON, выполняются команды, находящиеся между IF(802) и ELSE(803). Если условие выполнения находится в состоянии OFF, выполняются команды, находящиеся между ELSE(803) и IEND(804).	Блок программы/ Требуется	686
CONDITIONAL BLOCK BRANCHING	IF	802	Если операнд находится в состоянии ON, выполняются команды, находящиеся между IF(802) и ELSE(803). Если операнд находится в состоянии OFF, выполняются команды, находящиеся между ELSE(803) и IEND(804).	Блок программы/ Требуется	686
CONDITIONAL BLOCK BRANCHING (NOT)	IF	802	Если операнд находится в состоянии ON, выполняются команды, находящиеся между IF(802) и ELSE(803). Если операнд находится в состоянии OFF, выполняются команды, находящиеся между ELSE(803) и IEND(804).	Блок программы/ Требуется	686
CONDITIONAL BLOCK BRANCHING (ELSE)	ELSE	803	Когда команда ELSE(803) опускается, а операнд находится в состоянии ON, выполняются команды, находящиеся между IF(802) и IEND(804).	Блок программы/ Требуется	686
CONDITIONAL BLOCK BRANCHING END	IEND	804	Если операнд находится в состоянии OFF, выполняются команды, находящиеся только после IEND(804).	Блок программы/ Требуется	686
ONE CYCLE AND WAIT	WAIT	805	Если условие выполнения для команды WAIT(805) находится в состоянии ON, оставшиеся команды блока программы пропускаются.	Блок программы/ Требуется	693
ONE CYCLE AND WAIT	WAIT	805	Если операнд находится в состоянии OFF (ON для команды WAIT NOT(805)), оставшиеся команды блока программы пропускаются.  В следующем цикле ни один из блоков программы выполняться не будет, кроме команд, выполняемых при наступлении условия выполнения для WAIT(805) и WAIT NOT(805). Когда условие выполнения переходит в состояние ON (OFF для WAIT(805) NOT), выполняются команды от WAIT(805) или WAIT(805)NOT до конца программы.	Блок программы/ Требуется	693
ONE CYCLE AND WAIT (NOT)	WAIT NOT	805	Если операнд находится в состоянии OFF (ON для команды WAIT NOT(805)), оставшиеся команды блока программы пропускаются.  В следующем цикле ни один из блоков программы выполняться не будет, кроме команд, выполняемых при наступлении условия выполнения для WAIT(805) и WAIT NOT(805). Когда условие выполнения переходит в состояние ON (OFF для WAIT(805) NOT), выполняются команды от WAIT(805) или WAIT(805)NOT до конца программы.	Блок программы/ Требуется	693
TIMER WAIT	TIMW	813	Откладывает выполнение оставшегося блока программы до истечения заданного времени. Выполнение программы продолжается с команды, следующей за TIMW(813), после истечения времени таймера.	Блок программы/ Требуется	697
COUNTER WAIT	CNTW	814	Откладывает выполнение оставшегося блока программы до достижения заданного значения. Выполнение программы продолжается с команды, следующей за CNTW(814) после выполнения счетчиком отсчета.	Блок программы/ Требуется	700
HIGH-SPEED TIMER WAIT	TMHW	815	Откладывает выполнение оставшегося блока программы до истечения заданного времени. Выполнение программы продолжается с команды, следующей за TMHW(815), после истечения времени таймера.	Блок программы/ Требуется	703
LOOP	LOOP	809	Команда LOOP(809) определяет начало циклической программы.	Блок программы/ Требуется	59

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
LEND	LEND	810	Команда LEND(810) или LEND(810) NOT определяет конец циклической программы. При достижении команды LEND(810) или LEND(810) NOT программа возвращается к следующему выполнению предшествующей команды LOOP(809), до момента, когда бит операнда для команд LEND(810) или LEND(810) NOT переводится в состояние ON или OFF (соответственно), или до момента, когда условия выполнения для команды LEND(810) переводятся в состояние ON.	Блок программы/ Требуется	705
LEND	LEND	810	Если операнд для команды LEND(810) находится в состоянии OFF (или ON для команды LEND(810) NOT) выполнение цикла повторяется, начиная с команды, следующей за командой LOOP(809). Если операнд для команды LEND(810) находится в состоянии ON (или OFF для команды LEND(810) NOT), выполнение цикла прекращается и программа переходит к выполнению команды, следующей за LEND(810) или LEND(810) NOT.	Блок программы/ Требуется	705
LEND NOT	LEND NOT	810	Команда LEND(810) или LEND(810) NOT определяет конец циклической программы. При достижении команды LEND(810) или LEND(810) NOT программа возвращается к следующему выполнению предшествующей команды LOOP(809), до момента, когда операнд для команд LEND(810) или LEND(810) NOT переводится в состояние ON или OFF (соответственно), или до момента, когда условия выполнения для команды LEND(810) переводятся в состояние ON.	Блок программы/ Требуется	705

## 2-2-29 Команды обработки текстовых строк программы

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
MOV STRING	MOV\$ @MOV\$	664	Осуществляет передачу текстовой строки.	Вывод/ Требуется	710
CONCATENATE STRING	+\$ @+\$	656	Осуществляет присоединение одной текстовой строки к другой текстовой строке.	Вывод/ Требуется	712
GET STRING LEFT	LEFT\$ @LEFT\$	652	Осуществляет выбор и считывание определенного количества символов с начала текстовой строки.	Вывод/ Требуется	714
GET STRING RIGHT	RGHT\$ @RGHT\$	653	Осуществляет выбор и считывание определенного количества символов с конца текстовой строки.	Вывод/ Требуется	716
GET STRING MIDDLE	MID\$ @MID\$	654	Осуществляет выбор и считывание определенного количества символов с любой позиции текстовой строки.	Вывод/ Требуется	719
FIND IN STRING	FIND\$ @FIND\$	660	Осуществляет нахождение заданной текстовой строки в текстовой строке.	Вывод/ Требуется	721
STRING LENGTH	LEN\$ @LEN\$	650	Осуществляет вычисление длины текстовой строки.	Вывод/ Требуется	723
REPLACE IN STRING	RPLC\$ @RPLC\$	661	Осуществляет замену текстовой строки заданной текстовой строкой, начиная с определенной позиции.	Вывод/ Требуется	725
DELETE STRING	DEL\$ @DEL\$	658	Осуществляет удаление определенной текстовой строки из середины текстовой строки.	Вывод/ Требуется	727
EXCHANGE STRING	XCHG\$ @XCHG\$	665	Осуществляет замену определенной текстовой строки другой текстовой строкой.	Вывод/ Требуется	730
CLEAR STRING	CLR\$ @CLR\$	666	Осуществляет стирание всего содержания текстовой строки и заполнение ее нулями (NUL, 00 шестн.).	Вывод/ Требуется	732
INSERT INTO STRING	IN\$ @IN\$	657	Осуществляет ввод текстовой строки в середину текстовой строки.	Вывод/ Требуется	733

## 2-3 Алфавитный список команд согласно их мнемоническому изображению

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
String Comparison	LD AND OR + +\$ <>\$ <\$ <=\$ >\$ >=\$ 670 (=)\$ 671 (<>\$) 672 (<\$) 673 (<=\$) 674 (>\$) 675 (>=\$)		Команды сравнения текстовой строки (=, <>, <, <=, >, >=) осуществляют сравнение двух текстовых строк в ASCII кодах, начиная с начала. Если результат сравнения верен, для команд LOAD, AND, OR создаются условия выполнения ON.	LD: не требуется. AND, OR: требуется.	736

### 2-2-30 Команды управления задачами

Команда	Мнемоника	Код	Функция	Расположение/ Условие выполнения	Страница
TASK ON	TKON @TKON	820	Разрешает выполнения определенной задачи.	Вывод/ Требуется	741
TASK OFF	TKOF @TKOF	821	Осуществляет перевод определенной задачи в состояние ожидания.	Вывод/ Требуется	744

## 2-3 Алфавитный список команд согласно их мнемоническому изображению

A

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
ACOS	ARC COSINE	464	@ACOS	–	–	427
AND	AND	–	@AND	%AND	!AND	89
AND<	AND LESS THAN	310	–	–	–	175
AND<\$	AND STRING LESS THAN	672	–	–	–	175
AND<>	AND NOT EQUAL	305	–	–	–	175
AND <>\$	AND STRING NOT EQUAL	671	–	–	–	175
AND <>L	AND DOUBLE NOT EQUAL	306	–	–	–	175
AND <>S	AND SIGNED NOT EQUAL	307	–	–	–	175
AND <>SL	AND DOUBLE SIGNED NOT EQUAL	308	–	–	–	175
AND<L	AND DOUBLE LESS THAN	311	–	–	–	175
AND<S	AND SIGNED LESS THAN	312	–	–	–	175
AND <SL	AND DOUBLE SIGNED LESS THAN	313	–	–	–	175
AND=	AND EQUAL	300	–	–	–	175
AND=\$	AND STRING EQUALS	670	–	–	–	175
AND=L	AND DOUBLE EQUAL	301	–	–	–	175
AND=S	AND SIGNED EQUAL	302	–	–	–	175
AND =SL	AND DOUBLE SIGNED EQUAL	303	–	–	–	175
AND>	AND GREATER THAN	320	–	–	–	175
AND>\$	AND STRING GREATER THAN	674	–	–	–	175
AND>L	AND DOUBLE GREATER THAN	321	–	–	–	175
AND>S	AND SIGNED GREATER THAN	322	–	–	–	175
AND >SL	AND DOUBLE SIGNED GREATER THAN	323	–	–	–	175
AND LD	AND LOAD	–	–	–	–	95
AND NOT	AND NOT	–	–	–	!AND NOT	90
AND TST	AND BIT TEST	350	–	–	–	104

2-3 Алфавитный список команд согласно их мнемоническому изображению

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
AND TSTN	AND BIT TEST	351	–	–	–	104
AND≤	AND LESS THAN OR EQUAL	315	–	–	–	
AND≤\$	AND STRING LESS THAN OR EQUALS	673	–	–	–	
AND≤L	AND DOUBLE LESS THAN OR EQUAL	316	–	–	–	
AND≤S	AND SIGNED LESS THAN OR EQUAL	317	–	–	–	
AND≤SL	AND DOUBLE SIGNED LESS THAN OR EQUAL	318	–	–	–	
AND≥	AND GREATER THAN OR EQUAL	325	–	–	–	
AND≥\$	AND STRING GREATER THAN OR EQUALS	675	–	–	–	
AND≥L	AND DOUBLE GREATER THAN OR EQUAL	326	–	–	–	
AND≥S	AND SIGNED GREATER THAN OR EQUAL	327	–	–	–	
AND≥SL	AND DOUBLE SIGNED GREATER THAN OR EQUAL	328	–	–	–	
ANDL	DOUBLE LOGICAL AND	610	@ANDL	–	–	
ANDW	LOGICAL AND	034	@ANDW	–	–	
APR	ARITHMETIC PROCESS	069	@APR	–	–	
ASC	ASCII CONVERT	086	@ASC	–	–	
ASFT	ASYNCHRONOUS SHIFT REGISTER	017	@ASFT	–	–	
ASIN	ARC SINE	463	@ASIN	–	–	
ASL	ARITHMETIC SHIFT LEFT	025	@ASL	–	–	
ASLL	DOUBLE SHIFT LEFT	570	@ASLL	–	–	
ASR	ARITHMETIC SHIFT RIGHT	026	@ASR	–	–	
ASRL	DOUBLE SHIFT RIGHT	571	@ASRL	–	–	
ATAN	ARC TANGENT	465	@ATAN	–	–	
AVG	AVERAGE	195	–	–	–	

**В**

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
BAND	DEAD BAND CONTROL	681	©BAND	–	–	
BCD	BINARY-TO-BCD	024	@BCD	–	–	
BCDL	DOUBLE BINARY-TO-BCD	059	@BCDL	–	–	
BCDS	SIGNED BINARY-TO-BCD	471	@BCDS	–	–	
BCMP	UNSIGNED BLOCK COMPARE	068	@BCMP	–	–	
BCNT	BIT COUNTER	067	@BCNT	–	–	
BDSL	DOUBLE SIGNED BINARY-TO-BCD	473	@BDSL	–	–	
BEND	BLOCK PROGRAM END	801	–	–	–	
BIN	BCD-TO-BINARY	023	@BIN	–	–	
BINL	DOUBLE BCD-TO-DOUBLE BINARY	058	@BINL	–	–	
BINS	SIGNED BCD-TO-BINARY	470	©BINS	–	–	
BISL	DOUBLE SIGNED BCD-TO-BINARY	472	@BISL	–	–	
BPPS	BLOCK PROGRAM PAUSE	811	–	–	–	
BPRG	BLOCK PROGRAM BEGIN	096	–	–	–	

2-3 Алфавитный список команд согласно их мнемоническому изображению

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
BPRS	BLOCK PROGRAM RESTART	812	–	–	–	
BREAK	BREAK LOOP	514	–	–	–	
BSET	BLOCK SE	071	@BSET	–	–	

**C**

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
CADD	CALENDAR ADD	730	@CADD	–	–	
CJP	CONDITIONAL JUMP	510	–	–	–	
CJPN	CONDITIONAL JUMP	511	–	–	–	
CLC	CLEAR CARRY	041	@CLC	–	–	
CLI	CLEAR INTERRUPT	691	@CLI	–	–	
CLRS	CLEAR STRING	666	@CLRS	–	–	
CMND	DELIVER COMMAND	490	@CMND	–	–	
CMP	COMPARE	020	–	–	!CMP	
CMPL	DOUBLE COMPARE	060	–	–	–	
CNR	RESET TIMER/COUNTER	545	@CNR	–	–	
CNT	COUNTER	–	–	–	–	
CNTR	REVERSIBLE COUNTER	012	–	–	–	
CNTW	COUNTER WAIT	814	–	–	–	
COLL	DATA COLLECT	081	@COLL	–	–	
COLM	LINE TO COLUMN	064	@COLM	–	–	
COM	COMPLEMENT	029	–	–	–	
COML	DOUBLE COMPLEMENT	614	@COML	–	–	
COS	COSINE	461	@COS	–	–	
CPS	SIGNED BINARY COMPARE	114	–	–	!CPS	
CPSL	DOUBLE SIGNED BINARY COM PARE	115	–	–	–	
CSUB	CALENDAR SUBTRACT	731	@CSUB	–	–	

**D**

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
DATE	CLOCK ADJUSTMENT	735	©DATE	–	–	
DEG	RADIANS–TO DEGREES	459	@DEG	–	–	
DELS	DELETE STRING	658	@DELS	–	–	
DI	DISABLE INTERRUPTS	693	@DI	–	–	
DIFD	DIFFERENTIATE DOWN	014	–	–	! DIFD	
DIFU	DIFFERENTIATE UP	013	–	–	! DIFU	
DIM	DIMENSION RECORD TABLE	631	@DIM	–	–	
DIST	SINGLE WORD DISTRIBUTE	080	@DIST	–	–	
DMPX	DATA ENCODER	077	@DMPX	–	–	
DOWN	CONDITION OFF	522	–	–	–	

**E**

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
EI	ENABLE INTERRUPTS	694	–	–	–	
ELSE	ELSE	803	–	–	–	
EMBC	SELECT EM BANK	281	@EMBC	–	–	

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
END	END	001	–	–	–	
EXIT NOT (operand)	CONDITIONAL BLOCK EXIT NOT	806	–	–	–	
EXIT (input condition)	CONDITIONAL BLOCK EXIT	806	–	–	–	
EXIT (operand)	CONDITIONAL BLOCK EXIT	806	–	–	–	
EXP	EXPONENT	467	@EXP	–	–	

**F**

Мнемониче- ка	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
FAL	FAILURE ALARM	006	@FAL	–	–	
FALS	SEVERE FAILURE ALARM	007	–	–	–	
FCS	FRAME CHECKSUM	180	@FCS	–	–	
FDIV	FLOATING POINT DIVIDE	079	@FDIV	–	–	
FIFO	FIRST IN FIRST OUT	633	©FIFO	–	–	
FINDS	FIND IN STRING	660	@FIND\$	–	–	
FIX	FLOATING TO 16-BIT	450	@FIX	–	–	
FIXL	FLOATING TO 32-BIT	451	@FIXL	–	–	
FLT	16-BIT TO FLOATING	452	@FLT	–	–	
FTL	32-BIT TO FLOATING	453	@FTL	–	–	
FOR	FOR-NEXT LOOPS	512	–	–	–	
FPD	FAILURE POINT DETECTION	269	–	–	–	
FREAD	READ DATA FILE	700	@FREAD	–	–	
FWRIT	WRITE DATA FIL	701	@FWRIT	–	–	

**G**

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
GETR	GET RECORD NUMBER	636	@GETR	–	–	

**H**

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
HEX	ASCII TO HEX	162	@HEX	–	–	
HMS	SECONDS TO HOURS	066	@HMS	–	–	

**I**

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
IEND	IF END	804	–	–	–	
IF NOT (operand)	IF NOT	802	–	–	–	
IF (input condition)	IF	802	–	–	–	
IF (operand)	IF	802	–	–	–	
IL	INTERLOCK	002	–	–	–	
ILC	INTERLOCK CLEAR	003	–	–	–	
INSS	INSS	657	@INSS	–	–	
IORD	INTELLIGENT I/O READ	222	@IORD	–	–	

2-3 Алфавитный список команд согласно их мнемоническому изображению

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
IORF	I/O REFRESH	097	@IORF	–	–	
IOWR	INTELLIGENT I/O WRITE	223	@IOWR	–	–	

**J**

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
JME	JUMP END	005	–	–	–	
JMEO	MULTIPLE JUMP END		–	–	–	
JMP	JUMP	004	–	–	–	
JMPO	MULTIPLE JUMP	515	–	–	–	

**K**

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
KEEP	KEEP	011	–	–	! KEEP	

**L**

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
LD	LOAD	–	@LD	%LD	ILD	
LD<	LOAD LESS THAN	310	–	–	–	
LD<\$	LOAD STRING LESS THAN	672	–	–	–	
LD<>	LOAD NOT EQUAL	305	–	–	–	
LD<>\$	LOAD STRING NOT EQUAL	671	–	–	–	
LD<>L	LOAD DOUBLE NOT EQUAL	306	–	–	–	
LD<>S	LOAD SIGNED NOT EQUAL	307	–	–	–	
LD<>SL	LOAD DOUBLE SIGNED NOT EQUAL	308	–	–	–	
LD<L	LOAD DOUBLE LESS THAN	311	–	–	–	
LD<S	LOAD SIGNED LESS THAN	312	–	–	–	
LD<SL	LOAD DOUBLE SIGNED LESS THAN	313	–	–	–	
LD=	LOAD EQUAL	300	–	–	–	
LD=\$	LOAD STRING EQUALS	670	–	–	–	
LD=L	LOAD DOUBLE EQUAL	301	–	–	–	
LD=S	LOAD SIGNED EQUAL	302	–	–	–	
LD=SL	LOAD DOUBLE SIGNED EQUAL	303	–	–	–	
LD>	LOAD GREATER THAN	320	–	–	–	
LD>\$	LOAD STRING GREATER THAN	674	–	–	–	
LD>L	LOAD DOUBLE GREATER THAN	321	–	–	–	
LD>S	LOAD SIGNED GREATER THAN	322	–	–	–	
LD>SL	LOAD DOUBLE SIGNED GREATER THAN	323	–	–	–	
LD NOT	LOAD NOT	–	–	–	!LD NOT	
LD TST	LOAD BIT TEST	350	–	–	–	
LD TSTN	LOAD BIT TEST	351	–	–	–	
LD≤	LOAD LESS THAN OR EQUAL	315	–	–	–	



2-3 Алфавитный список команд согласно их мнемоническому изображению

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
LD≤\$	LOAD STRING LESS THAN OR EQUAL	673	–	–	–	
LD≤L	LOAD DOUBLE LESS THAN OR EQUAL	316	–	–	–	
LD≤S	LOAD SIGNED LESS THAN OR EQUAL	317	–	–	–	
LD≤SL	LOAD DOUBLE SIGNED LESS THAN OR EQUAL	318	–	–	–	
LD≥	LOAD GREATER THAN OR EQUAL	325	–	–	–	
LD≥\$	LOAD STRING GREATER THAN OR EQUALS	675	–	–	–	
LD≥L	LOAD DOUBLE GREATER THAN OR EQUAL	326	–	–	–	
LD≥S	LOAD SIGNED GREATER THAN OR EQUAL	327	–	–	–	
LD≥SL	LOAD DOUBLE SIGNED GREATER THAN OR EQUAL	328	–	–	–	
LEFT\$	GET STRING LEFT	652	@LEFT\$	–	–	
LENS	STRING LENGTH	650	@LENS	–	–	
LEND NOT (operand)	LOOP END NOT	810	–	–	–	
LEND (input condition)	LOOP END	810	–	–	–	
LEND (operand)	LOOP END	810	–	–	–	
LIFO	LAST IN FIRST OUT	634	@LIFO	–	–	
LINE	COLUMN TO LINE	063	©LINE	–	–	
LMT	LIMIT CONTROL	680	@LMT	–	–	
LOG	LOGARITHM	468	@LOG	–	–	
LOOP	LOOP	809	–	–	–	

**М**

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
MAX	FIND MAXIMUM	182	@MAX	–	–	
MCMP	MULTIPLE COMPARE	019	@MCMP	–	–	
MCRO	MACRO	099	@MCRO	–	–	
MID\$	GET STRING MIDDLE	654	@MID\$	–	–	
MIN	FIND MINIMUM	183	@MIN	–	–	
MLPX	DATA DECODER	076	@MLPX	–	–	
MOV	MOVE	021	@MOV	–	! MOV	
MOV\$	MOVE STRING	664	@MOV\$	–	–	
MOVB	MOVE BIT	082	@MOVB	–	–	
MOVD	MOVE DIGIT	083	@MOVD	–	–	
MOVL	DOUBLE MOVE	498	@MOVL	–	–	
MOVR	MOVE TO REGISTER	560	@MOVR	–	–	
MSG	DISPLAY MESSAGE	046	@MSG	–	–	
MSKR	READ INTERRUPT MASK	692	@MSKR	–	–	
MSKS	SET INTERRUPT MASK	690	@MSKS	–	–	
MTIM	MULTI-OUTPUT TIMER	543	–	–	–	
MVN	MOVE NOT	022	@MVN	–	–	
MVNL	DOUBLE MOVE NOT	499	@MVNL	–	–	

2-3 Алфавитный список команд согласно их мнемоническому изображению

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
MOVRW	MOVE TIMER/COUNTER PV TO REGISTER	561				

N

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
NASL	SHIFT N-BITS LEFT	580	@NASL	–	–	
NASR	SHIFT N-BITS RIGHT	581	@NASR	–	–	
NEG	2'S COMPLEMENT	160	@NEG	–	–	
NEGL	DOUBLE COMPLEMENT 2'S	161	@NEGL	–	–	
NEXT	FOR-NEXT LOOPS	513	–	–	–	
NOP	NO OPERATION	000	–	–	–	
NOT	NOT	520	–	–	–	
NSFL	SHIFT N-BIT DATA LEFT	578	@NSFL	–	–	
NSFR	SHIFT N-BIT DATA RIGHT	579	@NSFR	–	–	
NSLL	DOUBLE SHIFT N-BITS LEFT	582	@NSLL	–	–	
NSRL	DOUBLE SHIFT N-BITS RIGHT	583	@NSRL	–	–	

O

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
OR	OR	–	@OR	%OR	!OR	
OR<	OR LESS THAN	310	–	–	–	
OR<\$	OR STRING LESS THAN	672	–	–	–	
OR<>	OR NOT EQUAL	305	–	–	–	
OR<>\$	OR STRING NOT EQUAL	671	–	–	–	
OR<>L	OR DOUBLE NOT EQUAL	306	–	–	–	
OR<>S	OR SIGNED NOT EQUAL	307	–	–	–	
OR<>SL	OR DOUBLE SIGNED NOT EQUAL	308	–	–	–	
OR<L	OR DOUBLE LESS THAN	311	–	–	–	
OR<S	OR SIGNED LESS THAN	312	–	–	–	
OR<SL	OR DOUBLE SIGNED LESS THAN	313	–	–	–	
OR=	OR EQUAL	300	–	–	–	
OR=\$	OR STRING EQUALS	670	–	–	–	
OR=L	OR DOUBLE EQUAL	301	–	–	–	
OR=S	OR SIGNED EQUAL	302	–	–	–	
OR=SL	OR DOUBLE SIGNED EQUAL	303	–	–	–	
OR>	OR GREATER THAN	320	–	–	–	
OR>\$	OR STRING GREATER THAN	674	–	–	–	
OR>L	OR DOUBLE GREATER THAN	321	–	–	–	
OR>S	OR SIGNED GREATER THAN	322	–	–	–	
OR>SL	OR DOUBLE SIGNED GREATER THAN	323	–	–	–	
OR LD	OR LOAD	–	–	–	–	
OR NOT	OR NOT	–	–	–	!OR NOT	

2-3 Алфавитный список команд согласно их мнемоническому изображению

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
OR TST	OR BIT TEST	350	–	–	–	
OR TSTN	OR BIT TEST	351	–	–	–	
OR≤	OR LESS THAN OR EQUAL	315	–	–	–	
OR≤\$	OR STRING LESS THAN OR EQUALS	673	–	–	–	
OR≤L	OR DOUBLE LESS THAN OR EQUAL	316	–	–	–	
OR≤S	OR SIGNED LESS THAN OR EQUAL	317	–	–	–	
OR≤SL	OR DOUBLE SIGNED LESS THAN OR EQUAL	318	–	–	–	
OR≥	OR GREATER THAN OR EQUAL	325	–	–	–	
OR≥\$	OR STRING GREATER THAN OR EQUALS	675	–	–	–	
OR≥L	OR DOUBLE GREATER THAN OR EQUAL	326	–	–	–	
OR≥S	OR SIGNED GREATER THAN OR EQUAL	327	–	–	–	
OR≥SL	OR DOUBLE SIGNED GREATER THAN OR EQUAL	328	–	–	–	
ORW	LOGICAL OR	035	@ORW	–	–	
ORWL	DOUBLE LOGICAL OR	611	@ORWL	–	–	
OUT	OUTPUT	–	–	–	!OUT	
OUT NOT	OUTPUT NOT	–	–	–	!OUT NOT	144

**P**

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
PID	PID CONTROL	190	–	–	–	
PMCR	PROTOCOL MACRO	260	@PMCR	–	–	
PUSH	PUSH ONTO STACK	632	©PUSH	–	–	
PWR	EXPONENTIAL POWER	840	@PWR	–	–	

**R**

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
RAD	DEGREES TO RADIANS	458	@RAD	–	–	
RECV	NETWORK RECEIVE	098	@RECV	–	–	
RET	SUBROUTINE RETURN	093	–	–	–	
RGHT\$	GET STRING RIGHT	653	@RGHT\$	–	–	
RLNC	ROTATE LEFT WITHOUT CARRY	574	@RLNC	–	–	
RLNL	DOUBLE ROTATE LEFT WITHOUT CARRY	576	@RLNL	–	–	
ROL	ROTATE LEFT	027	@ROL	–	–	
ROLL	DOUBLE ROTATE LEFT	572	©ROLL	–	–	
ROOT	BCD SQUARE ROOT	072	©ROOT	–	–	
ROR	ROTATE RIGHT	028	©ROR	–	–	
RORL	DOUBLE ROTATE RIGHT	573	©RORL	–	–	
ROTB	BINARY ROOT	620	©ROTB	–	–	
RPLC\$	REPLACE IN STRING	661	@RPLC\$	–	–	

2-3 Алфавитный список команд согласно их мнемоническому изображению

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
RRNC	ROTATE RIGHT WITHOUT CARRY	575	©RRNC	–	–	
RRNL	DOUBLE ROTATE RIGHT WITHOUT CARRY	577	©RRNL	–	–	
RSET	RESET	–	©RSET	%RSET	!RSET	
RSTA	MULTIPLE BIT RESET	531	©RSTA	–	–	
RXD	RECEIVE	235	©RXD	–	–	

S

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
SBN	SUBROUTINE ENTRY	092	–	–	–	
SBS	SUBROUTINE CALL	091	©SBS	–	–	
SCL	SCALING	194	©SCL	–	–	
SCL2	SCALING 2	486	@SCL2	–	–	
SCL3	SCALING 3	487	@SCL3	–	–	
SDEC	7–SEGMENT DECODER	078	@SDEC	–	–	
SEC	HOURS TO SECONDS	065	@SEC	–	–	
SEND	NETWORK SEND	090	©SEND	–	–	
SET	SET	–	@SET	%SET	!SET	
SETA	MULTIPLE BIT SET	530	@SETA	–	–	
SETR	SET RECORD LOCATION	635	@SETR	–	–	
SFT	SHIFT REGISTER	010	–	–	–	
SFTR	REVERSIBLE SHIFT REGISTER	084	@SFTR	–	–	
SIGN	16–BIT TO 32–BIT SIGNED BINARY	600	©SIGN	–	–	
SIN	SINE	460	@SIN	–	–	
SLD	ONE DIGIT SHIFT LEFT	074	@SLD	–	–	
SNXT	STEP START	009	–	–	–	
SQRT	SQUARE ROOT	466	@SQRT	–	–	
SRCH	DATA SEARCH	181	@SRCH	–	–	
SRD	ONE DIGIT SHIFT RIGHT	075	@SRD	–	–	
SSET	SET STACK	630	@SSET	–	–	
STC	SET CARRY	040	@STC	–	–	
STEP	STEP DEFINE	008	–	–	–	
STUP	CHANGE SERIAL PORT SETUP	237	@STUP	–	–	
SUM	SUM	184	@SUM	–	–	
SWAP	SWAP BYTES	637	©SWAP	–	–	

T

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
TAN	TANGENT	462	@TAN	–	–	
TCMP	TABLE COMPARE	085	@TCMP	–	–	
TIM	TIMER	–	–	–	–	
TIMH	HIGH–SPEED TIMER	015	–	–	–	
TIML	LONG TIMER	542	–	–	–	
TIMW	TIMER WAIT	813	–	–	–	
TKOF	TASK OFF	821	@TKOF	–	–	
TKON	TASK ON	820	@TKON	–	–	
TMHH	ONE–MS TIMER	540	–	–	–	

2-3 Алфавитный список команд согласно их мнемоническому изображению

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
TMHW	HIGH-SPEED TIMER WAIT	815	–	–	–	
TRSM	TRACE MEMORY SAMPLING	045	–	–	–	
TTIM	ACCUMULATIVE TIMER	087	–	–	–	
TXD	TRANSMIT	236	@TXD	–	–	

U

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
UP	CONDITION ON	521	–	–	–	

W

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
WAIT NOT (operand)	ONE CYCLE AND WAIT NOT	805	–	–	–	
WAIT (input condition)	ONE CYCLE AND WAIT	805	–	–	–	
WAIT (operand)	ONE CYCLE AND WAIT	805	–	–	–	
WDT	EXTEND MAXIMUM CYCLE TIME	094	@WDT	–	–	
WSFT	WORD SHIFT	016	@WSFT	–	–	

X

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
XCGL	DOUBLE EXCHANGE DATA	562	@XCGL	–	–	
XCHG	DATA EXCHANGE	073	@XCHG	–	–	
XCHG\$	EXCHANGE STRING	665	@XCHG\$	–	–	
XFER	BLOCK TRANSFER	070	@XFER	–	–	
XFRB	MULTIPLE BIT TRANSFER	062	@XFRB	–	–	
XNRL	DOUBLE EXCLUSIVE NOR	613	@XNRL	–	–	
XNRW	EXCLUSIVE NOR	037	@XNRW	–	–	
XORL	DOUBLE EXCLUSIVE OR	612	@XORL	–	–	
XORW	EXCLUSIVE OR	036	@XORW	–	–	

Z

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
ZONE	DEAD ZONE CONTROL	682	©ZONE	–	–	545

СИМВОЛЫ

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
+	SIGNED BINARY ADD WITHOUT CARRY	400	@+	–	–	
+\$	CONCATENATE STRING	656	@+\$	–	–	
++	INCREMENT BINARY	590	@++	–	–	
++B	INCREMENT BCD	594	@++B	–	–	
++BL	DOUBLE INCREMENT BCD	595	@++BL	–	–	

2-3 Алфавитный список команд согласно их мнемоническому изображению

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
++L	DOUBLE INCREMENT BINARY	591	@++L	–	–	
+B	BCD ADD WITHOUT CARRY	404	@+B	–	–	
+BC	BCD ADD WITH CARRY	406	@+BC	–	–	
+BCL	DOUBLE BCD ADD WITH CARRY	407	@+BCL	–	–	
+BL	DOUBLE BCD ADD WITHOUT CARRY	405	@+BL	–	–	
+C	SIGNED BINARY ADD WITH CARRY	402	@+C	–	–	
+CL	DOUBLE SIGNED BINARY ADD WITH CARRY	403	@+CL	–	–	
+F	FLOATING-POINT ADD	454	@+F	–	–	
+L	DOUBLE SIGNED BINARY ADD WITHOUT CARRY	401	@+L	–	–	
–	SIGNED BINARY SUBTRACT WITHOUT CARRY	410	@ –	–	–	
--	DECREMENT BINARY	592	@ --	–	–	
--B	DECREMENT BCD	596	@ -- B	–	–	
--BL	DOUBLE DECREMENT BCD	597	@ -- BL	–	–	
--L	DOUBLE DECREMENT BINARY	593	@ --L	–	–	
-B	BCD SUBTRACT WITHOUT CARRY	414	@ – B	–	–	
-BC	BCD SUBTRACT WITH CARRY	416	@ – BC	–	–	
-BCL	DOUBLE BCD SUBTRACT WITH CARRY	417	@ – BCL	–	–	
-BL	DOUBLE BCD SUBTRACT WITHOUT CARRY	415	@ – BL	–	–	
-C	SIGNED BINARY SUBTRACT WITH CARRY	412	@ – C	–	–	
-CL	DOUBLE SIGNED BINARY SUBTRACT WITH CARRY	413	@ – CL	–	–	
-F	FLOATING-POINT SUBTRACT	455	@ – F	–	–	
*	SIGNED BINARY MULTIPLY	420	@*	–	–	
*B	BCD MULTIPLY	424	@*B	–	–	
*BL	DOUBLE BCD MULTIPLY	425	@*BL	–	–	
*F	FLOATING-POINT MULTIPLY	456	@*F	–	–	
*L	DOUBLE SIGNED BINARY MULTIPLY	421	@*L	–	–	
*U	UNSIGNED BINARY MULTIPLY	422	@*U	–	–	
*UL	DOUBLE UNSIGNED BINARY MULTIPLY	423	@*UL	–	–	
-L	DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	411	@-L	–	–	
/	SIGNED BINARY DIVIDE	430	@/	–	–	
/B	BCD DIVIDE	434	@/B	–	–	
/BL	DOUBLE BCD DIVIDE	435	@/BL	–	–	
/F	FLOATING-POINT DIVIDE	457	@/F	–	–	
/L	DOUBLE SIGNED BINARY DIVIDE	431	@/L	–	–	

2-4 Перечень команд по функциональному коду

Мнемоника	Команда	Код функции	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
/U	UNSIGNED BINARY DIVIDE	432	@/U	–	–	
/UL	DOUBLE UNSIGNED BINARY DIVIDE	433	@/UL	–	–	

2-4 Перечень команд по функциональному коду

Код	Мнемоника	Команда	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
–	LD	LOAD	@LD	%LD	ILD	
–	LDNOT	LOAD NOT	–	–	!LD NOT	
–	AND	AND	@AND	%AND	!AND	
–	AND NOT	AND NOT	–	–	!AND NOT	
–	OR	OR	@OR	%OR	!OR	
–	OR NOT	OR NOT	–	–	!OR NOT	
–	ANDLD	AND LOAD	–	–	–	
–	ORLD	OR LOAD	–	–	–	
–	OUT	OUTPUT	–	–	IOUT	
–	OUT NOT	OUTPUT NOT	–	–	!OUT NOT	
–	SET	SET	@SET	%SET	!SET	
–	RSET	RESET	@RSET	%RSET	!RSET	
–	TIM	TIMER	–	–	–	
–	CNT	COUNTER	–	–	–	
000	NOP	NO OPERATION	–	–	–	
001	END	END	–	–	–	
002	IL	INTERLOCK	–	–	–	
003	ILC	INTERLOCK CLEAR	–	–	–	
004	JMP	JUMP	–	–	–	
005	JME	JUMP END	–	–	–	
006	FAL	FAILURE ALARM	@FAL	–	–	
007	FALS	SEVERE FAILURE ALARM	–	–	–	
008	STEP	STEP DEFINE	–	–	–	
009	SNXT	STEP START	–	–	–	
010	SFT	SHIFT REGISTER	–	–	–	
011	KEEP	KEEP	–	–	! KEEP	
012	CNTR	REVERSIBLE COUNTER	–	–	–	
013	DIFU	DIFFERENTIATE UP	–	–	!DIFU	
014	DIFD	DIFFERENTIATE DOWN	–	–	!DIFD	
015	TIMH	HIGH-SPEED TIMER	–	–	–	
016	WSFT	WORD SHIFT	@WSFT	–	–	
017	ASFT	ASYNCHRONOUS SHIFT REGISTER	@ASFT	–	–	
019	MCMP	MULTIPLE COMPARE	@MCMP	–	–	
020	CMP	COMPARE	–	–	!CMP	
021	MOV	MOVE	@MOV	–	! MOV	
022	MVN	MOVE NOT	@MVN	–	–	
023	BIN	BCD-TO-BINARY	@BIN	–	–	
024	BCD	BINARY-TO-BCD	@BCD	–	–	
025	ASL	ARITHMETIC SHIFT LEFT	@ASL	–	–	
026	ASR	ARITHMETIC SHIFT RIGHT	@ASR	–	–	
027	ROL	ROTATE LEFT	@ROL	–	–	
028	ROR	ROTATE RIGHT	@ROR	–	–	
029	COM	COMPLEMENT	@COM	–	–	



Код	Мнемоника	Команда	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
034	ANDW	LOGICAL AND	@ANDW	–	–	
035	ORW	LOGICAL OR	@ORW	–	–	
036	XORW	EXCLUSIVE OR	@XORW	–	–	
037	XNRW	EXCLUSIVE NOR	@XNRW	–	–	
040	STC	SET CARRY	@STC	–	–	
041	CLC	CLEAR CARRY	@CLC	–	–	
045	TRSM	TRACE MEMORY SAMPLING	–	–	–	
046	MSG	DISPLAY MESSAGE	@MSG	–	–	
058	BINL	DOUBLE BCD-TO-DOUBLE BINARY	@BINL	–	–	
059	BCDL	DOUBLE BINARY-TO-BCD	@BCDL	–	–	
060	CMPL	DOUBLE COMPARE	–	–	–	
062	XFRB	MULTIPLE BIT TRANSFER	@XFRB	–	–	
063	LINE	COLUMN TO LINE	©LINE	–	–	
064	COLM	LINE TO COLUMN	@COLM	–	–	
065	SEC	HOURS TO SECONDS	@SEC	–	–	
066	HMS	SECONDS TO HOURS	@HMS	–	–	
067	BCNT	BIT COUNTER	@BCNT	–	–	
068	BCMP	UNSIGNED BLOCK COMPARE	@BCMP	–	–	
069	APR	ARITHMETIC PROCESS	@APR	–	–	
070	XFER	BLOCK TRANSFER	@XFER	–	–	
071	BSET	BLOCK SET	@BSET	–	–	
072	ROOT	BCD SQUARE ROOT	©ROOT	–	–	
073	XCHG	DATA EXCHANGE	@XCHG	–	–	
074	SLD	ONE DIGIT SHIFT LEFT	@SLD	–	–	
075	SRD	ONE DIGIT SHIFT RIGHT	@SRD	–	–	
076	MLPX	DATA DECODER	@MLPX	–	–	
077	DMPX	DATA ENCODER	@DMPX	–	–	
078	SDEC	7-SEGMENT DECODER	@SDEC	–	–	
079	FDIV	FLOATING POINT DIVIDE	@FDIV	–	–	
080	DIST	SINGLE WORD DISTRIBUTE	@DIST	–	–	
081	COLL	DATA COLLECT	@COLL	–	–	
082	MOVB	MOVE BIT	@MOVB	–	–	
083	MOVD	MOVE DIGIT	@MOVD	–	–	
084	SFTR	REVERSIBLE SHIFT REGISTER	@SFTR	–	–	
085	TCMP	TABLE COMPARE	@TCMP	–	–	
086	ASC	ASCII CONVERT	@ASC	–	–	
087	TTIM	ACCUMULATIVE TIMER	–	–	–	
090	SEND	NETWORK SEND	©SEND	–	–	
091	SBS	SUBROUTINE CALL	@SBS	–	–	
092	SBN	SUBROUTINE ENTRY	–	–	–	
093	RET	SUBROUTINE RETURN	–	–	–	
094	WDT	EXTEND MAXIMUM CYCLE TIME	@WDT	–	–	
096	BPRG	BLOCK PROGRAM BEGIN	–	–	–	
097	IORF	I/O REFRESH	@IORF	–	–	
098	RECV	NETWORK RECEIVE	@RECV	–	–	
099	MCRO	MACRO	@MCRO	–	–	
114	CPS	SIGNED BINARY COMPARE	–	–	!CPS	
115	CPSL	DOUBLE SIGNED BINARY COMPARE	–	–	–	
160	NEG	2'S COMPLEMENT	@NEG	–	–	

Код	Мнемоника	Команда	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
161	NEGL	DOUBLE 2'S COMPLEMENT	@NEGL	–	–	
162	HEX	ASCII TO HEX	@HEX	–	–	
180	FCS	FRAME CHECKSUM	@FCS	–	–	
181	SRCH	DATA SEARCH	@SRCH	–	–	
182	MAX	FIND MAXIMUM	@MAX	–	–	
183	MIN	FIND MINIMUM	@MIN	–	–	
184	SUM	SUM	@SUM	–	–	
190	PID	PID CONTROL	–	–	–	
194	SCL	SCALING	@SCL	–	–	
195	AVG	AVERAGE	–	–	–	
222	IORD	INTELLIGENT I/O READ	@IORD	–	–	
223	IOWR	INTELLIGENT I/O WRITE	@IOWR	–	–	
235	RXD	RECEIVE	@RXD	–	–	
236	TXD	TRANSMIT	@TXD	–	–	
237	STUP	CHANGE SERIAL PORT SETUP	@STUP	–	–	
260	PMCR	PROTOCOL MACRO	@PMCR	–	–	
269	FPD	FAILURE POINT DETECTION	–	–	–	
281	EMBC	SELECT EM BANK	@EMBC	–	–	
300	AND=	AND EQUAL	–	–	–	
300	LD=	LOAD EQUAL	–	–	–	
300	OR=	OR EQUAL	–	–	–	
301	AND=L	AND DOUBLE EQUAL	–	–	–	
301	LD=L	LOAD DOUBLE EQUAL	–	–	–	
301	OR=L	OR DOUBLE EQUAL	–	–	–	
302	AND=S	AND SIGNED EQUAL	–	–	–	
302	LD=S	LOAD SIGNED EQUAL	–	–	–	
302	OR=S	OR SIGNED EQUAL	–	–	–	
303	AND =SL	AND DOUBLE SIGNED EQUAL	–	–	–	
303	LD=SL	LOAD DOUBLE SIGNED EQUAL	–	–	–	
305	OR=SL	OR DOUBLE SIGNED EQUAL	–	–	–	
305	AND <>	AND NOT EQUAL	–	–	–	
305	LD <>	LOAD NOT EQUAL	–	–	–	
305	OR <>	OR NOT EQUAL	–	–	–	
306	AND <>L	AND DOUBLE NOT EQUAL	–	–	–	
306	LD<>L	LOAD DOUBLE NOT EQUAL	–	–	–	
306	OR<>L	OR DOUBLE NOT EQUAL	–	–	–	
307	AND <>S	AND SIGNED NOT EQUAL	–	–	–	
307	LD<>S	LOAD SIGNED NOT EQUAL	–	–	–	
307	OR<>S	OR SIGNED NOT EQUAL	–	–	–	
308	AND <>SL	AND DOUBLE SIGNED NOT EQUAL	–	–	–	
308	LD <>SL	LOAD DOUBLE SIGNED NOT EQUAL	–	–	–	
308	OR <>SL	OR DOUBLE SIGNED NOT EQUAL	–	–	–	
310	AND<	AND LESS THAN	–	–	–	
310	LD<	LOAD LESS THAN	–	–	–	
310	OR<	OR LESS THAN	–	–	–	
311	AND<L	AND DOUBLE LESS THAN	–	–	–	
311	LD<L	LOAD DOUBLE LESS THAN	–	–	–	
311	OR<L	OR DOUBLE LESS THAN	–	–	–	
312	AND<S	AND SIGNED LESS THAN	–	–	–	

2-4 Перечень команд по функциональному коду

Код	Мнемоника	Команда	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
312	LD<S	LOAD SIGNED LESS THAN	–	–	–	
312	OR<S	OR SIGNED LESS THAN	–	–	–	
313	AND <SL	AND DOUBLE SIGNED LESS THAN	–	–	–	
313	LD<SL	LOAD DOUBLE SIGNED LESS THAN	–	–	–	
313	OR<SL	OR DOUBLE SIGNED LESS THAN	–	–	–	
315	AND≤	AND LESS THAN OR EQUAL	–	–	–	
315	LD≤	LOAD LESS THAN OR EQUAL	–	–	–	
315	OR≤	OR LESS THAN OR EQUAL	–	–	–	
316	AND≤L	AND DOUBLE LESS THAN OR EQUAL	–	–	–	
316	LD≤L	LOAD DOUBLE LESS THAN OR EQUAL	–	–	–	
316	OR≤L	OR DOUBLE LESS THAN OR EQUAL	–	–	–	
317	AND≤S	AND SIGNED LESS THAN OR EQUAL	–	–	–	
317	LD≤S	LOAD SIGNED LESS THAN OR EQUAL	–	–	–	
317	OR≤S	OR SIGNED LESS THAN OR EQUAL	–	–	–	
318	AND≤SL	AND DOUBLE SIGNED LESS THAN OR EQUAL	–	–	–	
318	LD≤SL	LOAD DOUBLE SIGNED LESS THAN OR EQUAL	–	–	–	
318	OR≤SL	OR DOUBLE SIGNED LESS THAN OR EQUAL	–	–	–	
320	AND>	AND GREATER THAN	–	–	–	
320	LD>	LOAD GREATER THAN	–	–	–	
320	OR>	OR GREATER THAN	–	–	–	
321	AND>L	AND DOUBLE GREATER THAN	–	–	–	
321	LD>L	LOAD DOUBLE GREATER THAN	–	–	–	
321	OR>L	OR DOUBLE GREATER THAN	–	–	–	
322	AND>S	AND SIGNED GREATER THAN	–	–	–	
322	LD>S	LOAD SIGNED GREATER THAN	–	–	–	
322	OR>S	OR SIGNED GREATER THAN	–	–	–	
323	AND >SL	AND DOUBLE SIGNED GREATER THAN	–	–	–	
323	LD>SL	LOAD DOUBLE SIGNED GREATER THAN	–	–	–	
323	OR>SL	OR DOUBLE SIGNED GREATER THAN	–	–	–	
325	AND≥	AND GREATER THAN OR EQUAL	–	–	–	
325	LD≥	LOAD GREATER THAN OR EQUAL	–	–	–	
325	OR≥	OR GREATER THAN OR EQUAL	–	–	–	
326	AND≥L	AND DOUBLE GREATER THAN OR EQUAL	–	–	–	
326	LD≥L	LOAD DOUBLE GREATER THAN OR EQUAL	–	–	–	
326	OR≥L	OR DOUBLE GREATER THAN OR EQUAL	–	–	–	
327	AND≥S	AND SIGNED GREATER THAN OR EQUAL	–	–	–	
327	LD≥S	LOAD SIGNED GREATER THAN OR EQUAL	–	–	–	

Код	Мнемоника	Команда	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
327	OR $\geq$ S	OR SIGNED GREATER THAN OR EQUAL	–	–	–	
328	AND $\geq$ SL	AND DOUBLE SIGNED GREATER THAN OR EQUAL	–	–	–	
328	LD $\geq$ SL	LOAD DOUBLE SIGNED GREATER THAN OR EQUAL	–	–	–	
328	OR $\geq$ SL	OR DOUBLE SIGNED GREATER THAN OR EQUAL	–	–	–	
350	AND TST	AND BIT TEST	–	–	–	
350	LD TST	LOAD BIT TEST	–	–	–	
350	OR TST	OR BIT TEST	–	–	–	
351	AND TSTN	AND BIT TEST NOT	–	–	–	
351	LD TSTN	LOAD BIT TEST NOT	–	–	–	
351	OR TSTN	OR BIT TEST NOT	–	–	–	
400	+	SIGNED BINARY ADD WITHOUT CARRY	@+	–	–	
401	+L	DOUBLE SIGNED BINARY ADD WITHOUT CARRY	@+L	–	–	
402	+C	SIGNED BINARY ADD WITH CARRY	@+C	–	–	
403	+CL	DOUBLE SIGNED BINARY ADD WITH CARRY	@+CL	–	–	
404	+B	BCD ADD WITHOUT CARRY	@+B	–	–	
405	+BL	DOUBLE BCD ADD WITHOUT CARRY	@+BL	–	–	
406	+BC	BCD ADD WITH CARRY	@+BC	–	–	
407	+BCL	DOUBLE BCD ADD WITH CARRY	@+BCL	–	–	
410	–	SIGNED BINARY SUBTRACT WITHOUT CARRY	@–	–	–	
411	–L	DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	@–L	–	–	
412	–C	SIGNED BINARY SUBTRACT WITH CARRY	@–C	–	–	
413	–CL	DOUBLE SIGNED BINARY SUBTRACT WITH CARRY	@–CL	–	–	
414	–B	BCD SUBTRACT WITHOUT CARRY	@–B	–	–	
415	–BL	DOUBLE BCD SUBTRACT WITHOUT CARRY	@–BL	–	–	
416	–BC	BCD SUBTRACT WITH CARRY	@–BC	–	–	
417	–BCL	DOUBLE BCD SUBTRACT WITH CARRY	@–BCL	–	–	
420	*	SIGNED BINARY MULTIPLY	@*	–	–	
421	*L	DOUBLE SIGNED BINARY MULTIPLY	@*L	–	–	
422	*U	UNSIGNED BINARY MULTIPLY	@*U	–	–	
423	*UL	DOUBLE UNSIGNED BINARY MULTIPLY	@*UL	–	–	
424	*B	BCD MULTIPLY	@*B	–	–	
425	*BL	DOUBLE BCD MULTIPLY	@*BL	–	–	
430	/	SIGNED BINARY DIVIDE	@/	–	–	
431	/L	DOUBLE SIGNED BINARY DIVIDE	@/L	–	–	
432	/U	UNSIGNED BINARY DIVIDE	@/U	–	–	
433	/UL	DOUBLE UNSIGNED BINARY DIVIDE	@/UL	–	–	
434	/B	BCD DIVIDE	@/B	–	–	

Код	Мнемоника	Команда	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
435	/BL	DOUBLE BCD DIVIDE	@/BL	–	–	
450	FIX	FLOATING TO 16-BIT	@FIX	–	–	
451	FIXL	FLOATING TO 32-BIT	@FIXL	–	–	
452	FLT	16-BIT TO FLOATING	@FLT	–	–	
453	FLTL	32-BIT TO FLOATING	@FLTL	–	–	
454	+F	FLOATING-POINT ADD	@+F	–	–	
455	-F	FLOATING-POINT SUBTRACT	@-F	–	–	
456	F	FLOATING-POINT MULTIPLY	@F	–	–	
457	/F	FLOATING-POINT DIVIDE	@/F	–	–	
458	RAD	DEGREES TO RADIANS	@RAD	–	–	
459	DEG	RADIANS-TO DEGREES	@DEG	–	–	
460	SIN	SINE	@SIN	–	–	
461	COS	COSINE	@COS	–	–	
462	TAN	TANGENT	@TAN	–	–	
463	ASIN	ARC SINE	@ASIN	–	–	
464	ACOS	ARC COSINE	@ACOS	–	–	
465	ATAN	ARC TANGENT	@ATAN	–	–	
466	SQRT	SQUARE ROOT	@SQRT	–	–	
467	EXP	EXPONENT	@EXP	–	–	
468	LOG	LOGARITHM	@LOG	–	–	
470	BINS	SIGNED BCD-TO-BINARY	©BINS	–	–	
471	BCDS	SIGNED BINARY-TO-BCD	@BCDS	–	–	
472	BISL	DOUBLE SIGNED BCD-TO-BINARY	@BISL	–	–	
473	BDSL	DOUBLE SIGNED BINARY-TO-BCD	@BDSL	–	–	
486	SCL2	SCALING 2	@SCL2	–	–	
487	SCL3	SCALING 3	@SCL3	–	–	
490	CMND	DELIVER COMMAND	@CMND	–	–	
498	MOVL	DOUBLE MOVE	@MOVL	–	–	
499	MVNL	DOUBLE MOVE NOT	@MVNL	–	–	
510	CJP	CONDITIONAL JUMP	–	–	–	
511	CJPN	CONDITIONAL JUMP	–	–	–	
512	FOR	FOR-NEXT LOOPS	–	–	–	
513	NEXT	FOR-NEXT LOOPS	–	–	–	
514	BREAK	BREAK LOOP	–	–	–	
515	JMPO	MULTIPLE JUMP	–	–	–	
516	JMEO	MULTIPLE JUMP END	–	–	–	
520	NOT	NOT	–	–	–	
521	UP	CONDITION ON	–	–	–	
522	DOWN	CONDITION OFF	–	–	–	
530	SETA	MULTIPLE BIT SET	OSETA	–	–	
531	RSTA	MULTIPLE BIT RESET	©RSTA	–	–	
540	TMHH	ONE-MS TIMER	–	–	–	
542	TIML	LONG TIMER	–	–	–	
543	MTIM	MULTI-OUTPUT TIMER	–	–	–	
545	CNR	RESET TIMER/COUNTER	@CNR	–	–	
560	MOVR	MOVE TO REGISTER	@MOVR	–	–	
561	MOVRW	MOVE TIMER/COUNTER PV TO REGISTER	@MOVRW	–	–	
562	XCGL	DOUBLE DATA EXCHANGE	@XCGL	–	–	
570	ASLL	DOUBLE SHIFT LEFT	@ASLL	–	–	
571	ASRL	DOUBLE SHIFT RIGHT	@ASRL	–	–	

Код	Мнемоника	Команда	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
572	ROLL	DOUBLE ROTATE LEFT	©ROLL	–	–	
573	RORL	DOUBLE ROTATE RIGHT	©RORL	–	–	
574	RLNC	ROTATE LEFT WITHOUT CARRY	©RLNC	–	–	
575	RRNC	ROTATE RIGHT WITHOUT CARRY	©RRNC	–	–	
576	RLNL	DOUBLE ROTATE LEFT WITHOUT CARRY	©RLNL	–	–	
577	RRNL	DOUBLE ROTATE RIGHT WITHOUT CARRY	©RRNL	–	–	
578	NSFL	SHIFT N-BIT DATA LEFT	©NSFL	–	–	
579	NSFR	SHIFT N-BIT DATA RIGHT	©NSFR	–	–	
580	NASL	SHIFT N-BITS LEFT	©NASL	–	–	
581	NASR	SHIFT N-BITS RIGHT	©NASR	–	–	
582	NSLL	DOUBLE SHIFT N-BITS LEFT	©NSLL	–	–	
583	NSRL	DOUBLE SHIFT N-BITS RIGHT	©NSRL	–	–	
590	++	INCREMENT BINARY	©++	–	–	
591	++L	DOUBLE INCREMENT BINARY	@++L	–	–	
592	--	DECREMENT BINARY	@-	–	–	
593	-L	DOUBLE DECREMENT BINARY	@-L	–	–	
594	++B	INCREMENT BCD	@++B	–	–	
595	++BL	DOUBLE INCREMENT BCD	@++L	–	–	
596	-B	DECREMENT BCD	@-B	–	–	
597	-BL	DOUBLE DECREMENT BCD	@-BL	–	–	
600	SIGN	16-BIT TO 32-BIT SIGNED BINARY	©SIGN	–	–	
610	ANDL	DOUBLE LOGICAL AND	@ANDL	–	–	
611	ORWL	DOUBLE LOGICAL OR	@ORWL	–	–	
612	XORL	DOUBLE EXCLUSIVE OR	@XORL	–	–	
613	XNRL	DOUBLE EXCLUSIVE NOR	@XNRL	–	–	
614	COML	DOUBLE COMPLEMENT	@COML	–	–	
620	ROTB	BINARY ROOT	@ROTB	–	–	
630	SSET	SET STACK	@SSET	–	–	
631	DIM	DIMENSION RECORD TABLE	@DIM	–	–	
632	PUSH	PUSH ONTO STACK	©PUSH	–	–	
633	FIFO	FIRST IN FIRST OUT	©FIFO	–	–	
634	LIFO	LAST IN FIRST OUT	@LIFO	–	–	
635	SETR	SET RECORD LOCATION	@SETR	–	–	
636	GETR	GET RECORD NUMBER	@GETR	–	–	
637	SWAP	SWAP BYTES	©SWAP	–	–	
650	LENS	STRING LENGTH	@LENS	–	–	
652	LEFTS	GET STRING LEFT	@LEFTS	–	–	
653	RGHTS	GET STRING RIGHT	@RGHTS	–	–	
654	MIDS	GET STRING MIDDLE	@MIDS	–	–	
656	+\$	CONCATENATE STRING	@+\$	–	–	
657	INSS	INSS	@INSS	–	–	
658	DELS	DELETE STRING	@DELS	–	–	
660	FINDS	FIND IN STRING	@FINDS	–	–	
661	RPLCS	REPLACE IN STRING	@RPLCS	–	–	
664	MOV\$	MOV STRING	@MOV\$	–	–	
665	XCHG\$	EXCHANGE STRING	@XCHG\$	–	–	
666	CLR\$	CLEAR STRING	@CLR\$	–	–	
670	AND=\$	AND STRING EQUALS	–	–	–	
670	LD=\$	LOAD STRING EQUALS	–	–	–	

2-4 Перечень команд по функциональному коду

Код	Мнемоника	Команда	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
670	OR=\$	OR STRING EQUALS	–	–	–	
671	AND <>\$	AND STRING NOT EQUAL	–	–	–	
671	LD<>\$	LOAD STRING NOT EQUAL	–	–	–	
671	OR<>\$	OR STRING NOT EQUAL	–	–	–	
672	AND<\$	AND STRING LESS THAN	–	–	–	
672	LD<\$	LOAD STRING LESS THAN	–	–	–	
672	OR<\$	OR STRING LESS THAN	–	–	–	
673	AND≤i\$	AND STRING LESS THAN OR EQUALS	–	–	–	
673	LD≤\$	LOAD STRING LESS THAN OR EQUAL	–	–	–	
673	OR≤\$	OR STRING LESS THAN OR EQUALS	–	–	–	
674	AND>\$	AND STRING GREATER THAN	–	–	–	
674	LD>\$	LOAD STRING GREATER THAN	–	–	–	
674	OR>\$	OR STRING GREATER THAN	–	–	–	
675	AND≥\$	AND STRING GREATER THAN OR EQUALS	–	–	–	
675	LD≥\$	LOAD STRING GREATER THAN OR EQUALS	–	–	–	
675	OR≥\$	OR STRING GREATER THAN OR EQUALS	–	–	–	
680	LMT	LIMIT CONTROL	@LMT	–	–	
681	BAND	DEAD BAND CONTROL	©BAND	–	–	
682	ZONE	DEAD ZONE CONTROL	©ZONE	–	–	
690	MSKS	SET INTERRUPT MASK	@MSKS	–	–	
691	CLI	CLEAR INTERRUPT	@CLI	–	–	
692	MSKR	READ INTERRUPT MASK	@MSKR	–	–	
693	DI	DISABLE INTERRUPTS	@DI	–	–	
694	EI	ENABLE INTERRUPTS	–	–	–	
700	FREAD	READ DATA FILE	@FREAD	–	–	
701	FWRIT	WRITE DATA FILE	@FWRIT	–	–	
730	CADD	CALENDAR ADD	@CADD	–	–	
731	CSUB	CALENDAR SUBTRACT	@CSUB	–	–	
735	DATE	CLOCK ADJUSTMENT	©DATE	–	–	
801	BEND	BLOCK PROGRAM END	–	–	–	
802	IF	CONDITIONAL BRANCHING BLOCK	–	–	–	
802	IF	CONDITIONAL BRANCHING BLOCK	–	–	–	
802	IF NOT	CONDITIONAL BRANCHING BLOCK NOT	–	–	–	
803	ELSE	ELSE	–	–	–	
804	IEND	IF END	–	–	–	
805	WAIT	ONE CYCLE AND WAIT	–	–	–	
805	WAIT	ONE CYCLE AND WAIT	–	–	–	
805	WAIT NOT	ONE CYCLE AND WAIT NOT	–	–	–	
806	EXIT	CONDITIONAL BLOCK EXIT	–	–	–	
806	EXIT	CONDITIONAL BLOCK EXIT	–	–	–	
806	EXIT NOT	CONDITIONAL BLOCK EXIT NOT	–	–	–	
809	LOOP	LOOP	–	–	–	
810	LEND	LOOP END	–	–	–	
810	LEND	LOOP END	–	–	–	



2-4 Перечень команд по функциональному коду

---

Код	Мнемоника	Команда	Дифференцирование вверх	Дифференцирование вниз	Немедленная регенерация	Страница
810	LEND NOT	LOOP END NOT	–	–	–	
811	BPPS	BLOCK PROGRAM PAUSE	–	–	–	
812	BPRS	BLOCK PROGRAM RESTART	–	–	–	
813	TIMW	TIMER WAIT	–	–	–	
814	CNTW	COUNTER WAIT	–	–	–	
815	TMHW	HIGH-SPEED TIMER WAIT	–	–	–	
820	TKON	TASK ON	@TKON	–	–	
821	TKOF	TASK OFF	@TKOF	–	–	
840	PWR	EXPONENTIAL POWER	@PWR	–	–	

---

## Глава 3 Команды

---

*В настоящей главе приводится описание команд, которые могут применяться при программировании работы Программируемых контроллеров серии CSI. Описание команд дается в последовательности согласно их функциональному назначению и классификации в Разделе 2 «Краткое описание команд».*

### 3-1 Система обозначений и расположение описания команды

Описание команд приводится по группам, согласно выполняемым функциям. Для справки обратитесь к разделу 2-3, содержащему алфавитный список команд согласно их мнемоническому изображению, в котором указаны страницы, на которых приводится описание каждой из команд.

Описание каждой из команд организовано таким образом:

#### Наименование и мнемоническое изображение

Заголовок каждого из разделов содержит наименование команды, сопровождаемое мнемоническим изображением и функциональным кодом, заключенным в скобки. Например: MOVE BIT: MOV B (082)

#### Назначение

Основное назначение команды приводится сразу после заголовка раздела

#### Символ релейно-контактной схемы и наименование операнда

Символ релейно-контактной схемы используемый в СХ-программаторе приводится ниже, применительно к случаю использования команды MOVE BIT. Наименование каждого из операндов также сопровождается символами релейно-контактной схемы.

—	MOV B(082)	
	S	S: Исходное слово или данные
	C	C: Контрольное слово
	D	D: Слово назначения

#### Модификации

Модификации, которые могут использоваться для управления выполнением команды при особых условиях, указываются в мнемонической форме. Модификации, которые не применяются с определенной командой, сопровождаются надписью «не поддерживается».

- Выполняется в каждом цикле при условии выполнения в состоянии ON: команда выполняется в течение всего времени, пока условие выполнения находится в состоянии ON.
- Выполняется один раз при дифференцировании вверх: команда выполняется в следующем цикле только после перехода условия выполнения из состояния OFF в состояние ON.
- Выполняется один раз при дифференцировании вниз: команда выполняется в следующем цикле только после перехода условия выполнения из состояния ON в состояние OFF.
- Выполняется всегда: команда не требует условий выполнения и выполняется в каждом из циклов.
- Создает условие выполнения: команда выполняется в каждом из циклов для создания условия выполнения другой команды

#### Указание немедленной регенерации

Немедленная регенерация может указываться в некоторых командах для выполнения регенерации ввода/вывода непосредственно после выполнения команды. В случае, когда немедленная регенерация поддерживается, указание о регенерации дается в мнемонической форме. Когда немедленная регенерация командой не поддерживается, это указывается надписью «не поддерживается».

Модификации	Выполняется в каждом цикле для состояния ON.	MOV B(082)
	Выполняется один раз для дифференцирования вверх	@ MOV B(082)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

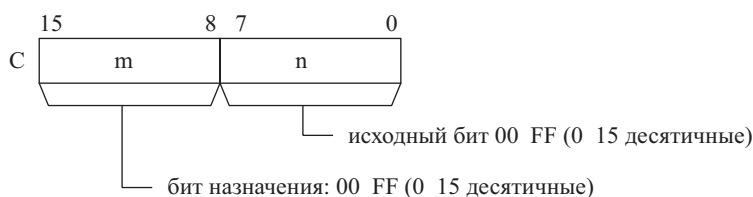
#### Используемые области программы

Области программы, в которых может использоваться команда, указываются. Надпись «Да» отмечает области, где команда может применяться.

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

Там, где это необходимо указываются значения слов и битов, используемых в отдельных операндах, например контрольные слова



### Спецификации операндов

Адреса областей памяти, которые могут использоваться для операндов, перечисляются в таблице, подобной приводимой ниже. Буквы, используемые в заголовках колонок, аналогичны применяемым символам в релейно-контактной схеме. Обозначение «-» указывается для индикации областей, которые не могут использоваться для операндов.

### Описание

Приводится функциональное назначение команды и операндов, используемых в команде.

### Флаги

Таблица флагов указывает состояния флагов непосредственно после выполнения команды. Флаги, которые не приводятся в таблице, при выполнении команды свое состояние не изменяют. Надпись «OFF» означает, что флаг переводится в состояние OFF непосредственно после выполнения команды, независимо от результатов выполнения команды.

### Предосторожности

Приводятся особые меры предосторожности, требуемые при использовании команды. Непременно соблюдайте указываемые меры предосторожности.

### Пример

Для более подробного пояснения функционального назначения команды приводится пример использования команды с конкретным операндом.

### Константы

Константы, вводимые в операнды, приводятся, как указано ниже.

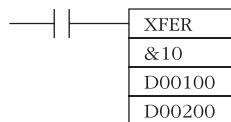
#### Описание и спецификации операндов

Операнды, содержащие строки битов (обычно вводятся в шестнадцатеричном коде):

- Для операндов, содержащих строки битов, указывается только форма в виде шестнадцатеричных данных, т.е. в качестве операнда S команды MOV(021) указывается только «#0000 #FFFF». Тем не менее, в CX-программаторе строка битов может вводиться в виде десятичного числа, с использованием префикса &.
- Операнды, содержащие числовые значения (обычно вводятся в виде десятичного числа, включая номера переходов):  
Для операндов, содержащих числовые значения, указывается обе формы - в десятичном коде и в шестнадцатеричном коде, т.е. в качестве операнда N команды XFER(070) указывается «#0000 #FFFF» и «& 0 65535».
- Операнды, содержащие контрольные числа, (кроме номеров переходов):  
Для операндов, содержащих контрольные числа, указывается форма в виде десятичных данных, т.е. в качестве операнда N команды SBS(091) указывается и «0 1023».

#### Примеры

В примерах константы приводятся с обозначением, используемым в CX-программаторе, т.е. операнды, содержащие числовые значения приводятся в виде десятичного числа с префиксом &, как показано на следующем рисунке.



Методы ввода констант при помощи Устройств программирования приводятся в следующей ниже таблице.

Операнд	CX-программатор	Пульт программирования
Операнды, содержащие строку битов, (обычно вводятся в шестнадцатеричном коде).	Вводите в десятичном коде с префиксом & или вводите в шестнадцатеричном коде с префиксом # (См. примечание.)	Для ввода шестнадцатеричных данных по умолчанию с префиксом # должны нажиматься клавиши Cont/# . Для изменения значений между шестнадцатеричными значениями (с префиксом #), десятичными со знаком (с +/-) и десятичными без знака (с &) значениями используется клавиша CHG.
Операнды, содержащие числовые значения (обычно вводятся в десятичном коде).		
Операнды, содержащие контрольные числа, (кроме номеров переходов).	Вводите в десятичном коде с префиксом # (См. примечание.)	Вводите в десятичном коде. Когда автоматически добавляется префикс & клавиша CHG используется для изменения значений между десятичными со знаком (с +/-), шестнадцатеричными (с префиксом #), и десятичными значениями без знака (с &).

**Примечание:** Когда операнды вводятся с помощью CX-программатора, диапазон вводимых значений выводится на дисплей с соответствующими префиксами.

### Флаги состояний

В данном параграфе в качестве флагов состояний используются обозначения Пульта программирования. При использовании CX-программатора, флаги состояний предварительно регистрируются как глобальные символы с буквой «P\_» впереди наименования символа.

Флаг	Пульт программирования	Обозначение CX-программатора
Флаг ошибки	ER	P_ER
Флаг ошибки доступа	AER	P_AER
Флаг переноса	CY	P_CY
Флаг более чем	>	P_GT
Флаг равенства	=	P_EQ
Флаг менее чем	<	P_LT
Флаг отрицательного значения	N	P_N
Флаг переполнения	OF	P_OF
Флаг отрицательного переполнения	UF	P_UF
Флаг более чем или равно	>=	P_GE
Флаг неравенства	<>	P_NE
Флаг менее чем или равенства	<=	P_LE
Флаг всегда ON	ON	P_On
Флаг всегда OFF	OFF	P_Off

### 3-2 Дополнительные возможности для версии 1 Модуля центрального процессора серии CS1

Для версии 1 Модулей центрального процессора приводимые ниже команды обладают дополнительными возможностями. Для детального ознакомления обратитесь к указанным ниже страницам Руководства. Функции, поддерживаемые только версией 1 Модулей центрального процессора, отмечены надписью «только –EV1».

Наименование	Мнемоника	Код	Функция	Дополнительные возможности	Страница
READ DATA FILE	FREAD	700	Поддерживается формат CSV и текстовый формат (.txt). Ранее поддерживались только двоичные данные.	Содержание контрольных данных изменено путем добавления формата данных, наличием возврата каретки, и указанием позиции возврата каретки.	
WRITE DATA FILE	FWRIT	701			
DELIVER COMMAND	CMND	490	Теперь Модуль центрального процессора может самостоятельно передавать команды FINS. (Ранее передача команд не поддерживалась)	Добавлена возможность передачи команд FINS в Модуль центрального процессора путем выполнения команды CMND(490).	

## 3-3 Команды последовательного ввода

## 3-3-1 LOAD: LD

**Назначение**

Обозначает запуск алгоритма и создает Условие выполнения ON/OFF, базируясь на состоянии бита (ON/OFF), указанного в операнде.

**Символ релейно-контактной схемы****Модификации**

Модификации	Осуществляет перезапуск алгоритма и создает состояние ON в каждом цикле, когда бит-операнд находится в состоянии ON	LD
	Осуществляет перезапуск алгоритма и создает состояние ON один раз при дифференцировании вверх	@LD
	Осуществляет перезапуск алгоритма и создает состояние ON один раз при дифференцировании вниз	%LD
Указание немедленной регенерации	!LD	
Комбинированные модификации	Осуществляет регенерацию бита ввода, перезапускает алгоритм и создает состояние ON один раз при дифференцировании вверх	!@LD
	Осуществляет регенерацию бита ввода, перезапускает алгоритм и создает состояние ON один раз при дифференцировании вниз	!%LD

**Применяемые области программы**

Область блоков	Область шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	LD бит-операнд
Область ввода/вывода (область CIO)	CIO 000000...CIO 614315
Рабочая область	W00000...W51115
Область удержания бита (Holding bit area)	H00000...H51115
Область вспомогательных битов	A00000...A95915
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область флага задачи	TK0000...TK0031
Флаги состояния	ER, CY, N, OF, UF, >, =, <, >=, <>, <=, A1, A0
Тактовые импульсы	0.02 сек, 0.1 сек, 0.2 сек, 1 сек, 1 мин.
Область TR	TR0...TR15
Область DM	—
Область EM, не содержащая банков	—
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	—
Регистры данных	—
Индексные регистры	—

Область	LD бит-операнд
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0 DR15, IR0 IR15 ,IR0+(++)...IR15(++) ,(- -)IR0...,-(- -)IR15

### Описание

Команда LD используется для первого от линии шины нормально открытого бита или для первого нормально открытого бита логического блока. Если немедленная регенерация не указывается, осуществляется чтение указанного бита в памяти ввода/вывода. Если немедленная регенерация указывается, осуществляется чтение и последующее использование состояния входной клеммы (ввода) Базового модуля ввода/вывода.

Команда LD используется в следующих обстоятельствах в качестве команды для индикации запуска алгоритма.

- Когда команда непосредственно подсоединена к линии шины.
- Когда логические блоки соединены командами AND LD или OR LD, т.е. в начале логического блока.

Команды AND LD и OR LD используются для последовательного или параллельного соединения логических блоков, начинающихся с команд LD или LD NOT.

По меньшей мере, одна команда LOAD или LOAD NOT требуется для условия выполнения, когда относящиеся к выводу команды не могут соединяться непосредственно к линии шины. Если команды LOAD или LOAD NOT отсутствуют, при проверке программы Периферийным устройством определяется ошибка программы.

Когда логические блоки соединяются с помощью команд AND LOAD или OR LOAD общее количество команд AND LOAD/OR LOAD должно соответствовать количеству команд LOAD/LOAD NOT минус 1. Если эти количества не совпадают, определяется ошибка команды. Для детального ознакомления обратитесь к разделам 3-3-7 AND LD и 3-3-8 OR LD.

### Флаги

Не существует флагов, на которые эта команда оказывает влияние.

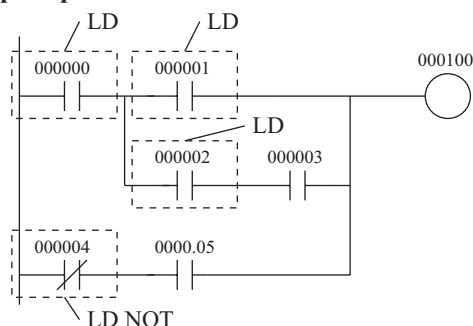
### Меры предосторожности

Для команды LD может задаваться дифференцирование вверх (@) или дифференцирование вниз (%). Если указывается дифференцирование вверх (@), условие выполнения переводится в состояние ON на время одного цикла только тогда, когда бит-операнд переходит из состояния OFF в состояние ON. Если указывается дифференцирование вниз (%), условие выполнения переводится в состояние ON на время одного цикла только тогда, когда бит-операнд переходит из состояния ON в состояние OFF.

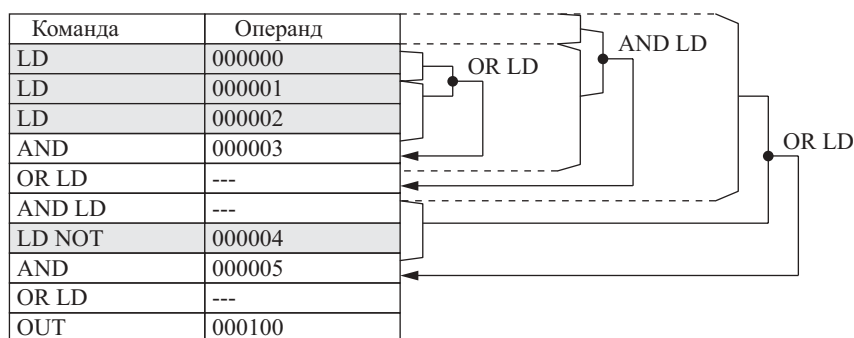
Для команды LD может задаваться немедленная регенерация (!). Команда немедленной регенерации осуществляет обновление состояния бита ввода непосредственно перед выполнением команды для Базовых модулей ввода (однако, не для Базовых модулей ввода, находящихся в Slave панелях и не для Многоточечных модулей ввода C200H группы 2).

Для команды LD можно объединять немедленную регенерацию и дифференцирование вверх или вниз (!@ или !%). При указании любой из этих команд ввод Базового модуля ввода подвергается регенерации непосредственно перед выполнением команды, и условие выполнения переводится в состояние ON на один цикл после перехода состояния бита из состояния OFF в состояние ON или наоборот.

### Пример







### 3-3-2 LOAD NOT: LD NOT

#### Назначение

Обозначает запуск алгоритма и создает условие выполнения ON/OFF, базируясь на инвертированное состояние (ON/OFF) бита, указанного в операнде.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Осуществляет перезапуск алгоритма и создает состояние ON в каждом цикле, когда бит-операнд находится в состоянии OFF.	LD NOT
	Осуществляет перезапуск алгоритма и создает состояние ON один раз при дифференцировании вверх	Не поддерживается.
	Осуществляет перезапуск алгоритма и создает состояние ON один раз при дифференцировании вниз	Не поддерживается.
Указание немедленной регенерации	!LD NOT	
Комбинированные модификации	Осуществляет регенерацию бита ввода, перезапускает алгоритм и создает состояние ON один раз при дифференцировании вверх	Не поддерживается
	Осуществляет регенерацию бита ввода, перезапускает алгоритм и создает состояние ON один раз при дифференцировании вниз	Не поддерживается

#### Применяемые области программы

Область блоков	Область шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Спецификации операндов

Область	LD NOT бит-операнд
Область ввода/вывода (область CIO)	CIO 000000...CIO 614315
Рабочая область	W00000...W51115
Область удержания бита (Holding bit area)	H00000...H51115
Область вспомогательных битов	A00000...A95915
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область флагов задачи	TK0000...TK0031
Флаги состояния	ER, CY, N, OF, UF, >, =, <, >=, <=, ON, OFF, AER
Тактовые импульсы	0.02 сек, 0.1 сек, 0.2 сек, 1 сек, 1 мин.
Область TR	—
Область DM	—
Область EM, не содержащая банков	—

Область	LD NOT бит-операнд
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0 DR15, IR0 IR15 ,IR0+(++)...IR15+(++) ,(- -)IR0...,(- -)IR15

### Описание

Команда LD NOT используется для первого от линии шины нормально закрытого бита или для первого нормально закрытого бита логического блока. Если немедленная регенерация не указывается, осуществляется чтение указанного бита в памяти ввода/вывода и изменение его состояния на противоположное. Если немедленная регенерация указывается, осуществляется чтение реверсирование состояния входной клеммы (ввода) Базового модуля ввода/вывода и последующее использование этого состояния.

Команда LD NOT используется в следующих обстоятельствах в качестве команды для индикации запуска алгоритма.

- Когда команда непосредственно подсоединена к линии шины.
- Когда логические блоки соединены командами AND LD или OR LD. (Используется в начале логического блока.)

Команды AND LD и OR LD используются для последовательного или параллельного соединения логических блоков, начинающихся с команд LD или LD NOT.

По меньшей мере, одна команда LOAD или LOAD NOT требуется для условия выполнения, когда относящиеся к выводу команды не могут соединяться непосредственно к линии шины. Если команды LOAD или LOAD NOT отсутствуют, при проверке программы Периферийным устройством определяется ошибка программы.

Когда логические блоки соединяются с помощью команд AND LOAD или OR LOAD общее количество команд AND LOAD/OR LOAD должно соответствовать количеству команд LOAD/LOAD NOT минус 1. Если эти количества не совпадают, определяется ошибка команды.

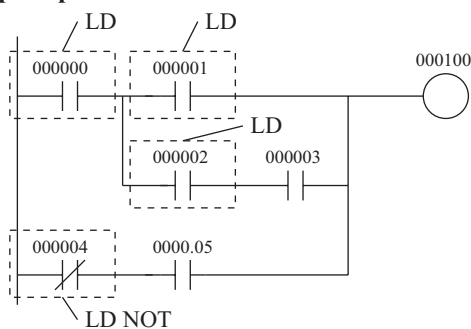
### Флаги

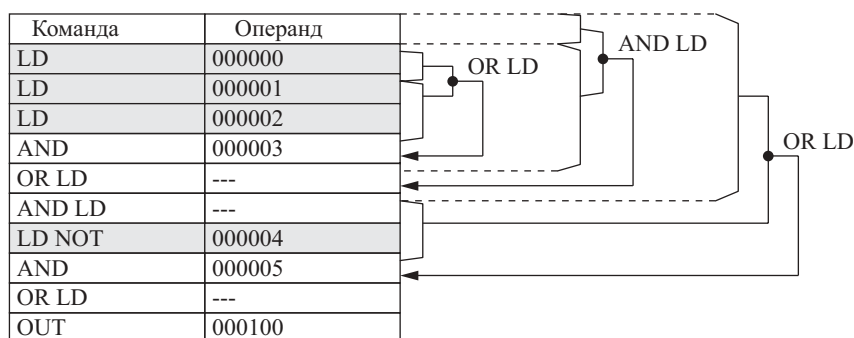
Не существует флагов, на которые эта команда оказывает влияние.

### Меры предосторожности

Для команды LD NOT может задаваться немедленная регенерация (!). Команда немедленной регенерации осуществляет обновление состояния бита ввода непосредственно перед выполнением команды для Базовых модулей ввода (однако, не для Базовых модулей ввода, находящихся в Slave панелях и не для Многоточечных модулей ввода C200H группы 2).

### Пример





### 3-3-3 AND: AND

#### Назначение

Выполняет логическую операцию «И» с состоянием бита указанного операнда и текущим условием выполнения.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Создает состояние ON в каждом цикле, когда результатом выполнения команды AND является состояние ON.	AND
	Создает состояние ON один раз при дифференцировании вверх.	@AND
	Создает состояние ON один раз при дифференцировании вниз.	%AND.
Указание немедленной регенерации	!AND	
Комбинированные модификации	Осуществляет регенерацию бита ввода и создает состояние ON один раз при дифференцировании вверх.	!@AND
	Осуществляет регенерацию бита ввода и создает состояние ON один раз при дифференцировании вниз.	!%AND

#### Применяемые области программы

Область блоков	Область шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Спецификации операндов

Область	AND бит-операнд
Область ввода/вывода (область CIO)	CIO 000000...CIO 614315
Рабочая область	W00000...W51115
Область удержания бита (Holding bit area)	H00000...H51115
Область вспомогательных битов	A00000...A95915
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область флагов задачи	TK0000...TK0031
Флаги состояния	ER, CY, N, OF, UF, >, =, <, >=, <=, ON, OFF, AER
Тактовые импульсы	0.02 сек, 0.1 сек, 0.2 сек, 1 сек, 1 мин.
Область TR	—
Область DM	—
Область EM, не содержащая банков	—
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—

Область	AND бит-операнд
Константы	–
Регистры данных	–
Индексные регистры	–
Косвенная адресация с использованием индексных регистров	,IR0...,IR15-2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0 DR15, IR0 IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-(-)IR15

**Описание**

Команда AND используется для нормально открытого бита, подключенного последовательно. Команда AND не может непосредственно подсоединяться к линии шины и не может использоваться в начале логического блока. Если немедленная регенерация не указывается, осуществляется чтение указанного бита в памяти ввода/вывода. Если немедленная регенерация указывается, осуществляется чтение состояния входной клеммы (ввода) Базового модуля ввода/вывода.

**Флаги**

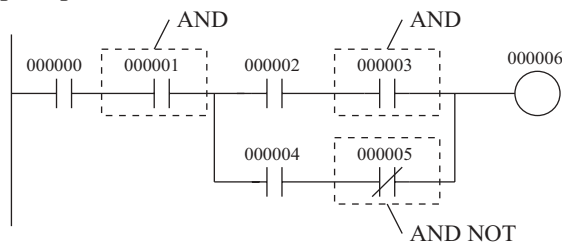
Не существует флагов, на которые эта команда оказывает влияние.

**Меры предосторожности**

Для команды AND может задаваться дифференцирование вверх (@) или дифференцирование вниз (%). Если указывается дифференцирование вверх (@), условие выполнения переводится в состояние ON на время одного цикла только тогда, когда бит-операнд переходит из состояния OFF в состояние ON. Если указывается дифференцирование вниз (%), условие выполнения переводится в состояние ON на время одного цикла только тогда, когда бит-операнд переходит из состояния ON в состояние OFF.

Для команды AND может задаваться немедленная регенерация (!). Команда немедленной регенерации осуществляет обновление состояния бита ввода непосредственно перед выполнением команды для Базовых модулей ввода (однако, не для Базовых модулей ввода, находящихся в Slave панелях и не для Многоточечных модулей ввода C200H группы 2).

Для команды AND можно объединять немедленную регенерацию и дифференцирование вверх или вниз (!@ или !%). При указании любой из этих команд ввод Базового модуля ввода подвергается регенерации непосредственно перед выполнением команды, и условие выполнения переводится в состояние ON на один цикл после перехода состояния бита из состояния OFF в состояние ON или наоборот.

**Пример****3-3-4 AND NOT: AND NOT****Назначение**

Выполняет логическую операцию «И» с инвертированным состоянием бита указанного операнда и текущим условием выполнения.

**Символ релейно-контактной схемы****Модификации**

Модификации	Создает состояние ON в каждом цикле, когда результатом выполнения команды AND NOT является состояние ON.	AND NOT
	Создает состояние ON один раз при дифференцировании вверх.	Не поддерживается.
	Создает состояние ON один раз при дифференцировании вниз.	Не поддерживается.

Указание немедленной регенерации	!AND NOT	
Комбинированные модификации	Осуществляет регенерацию бита ввода и создает состояние ON один раз при дифференцировании вверх.	Не поддерживается.
	Осуществляет регенерацию бита ввода и создает состояние ON один раз при дифференцировании вниз.	Не поддерживается.

### Применяемые области программы

Область блоков	Область шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

### Спецификации операндов

Область	AND NOT бит-операнд
Область ввода/вывода (область CIO)	CIO 000000...CIO 614315
Рабочая область	W00000...W51115
Область удержания бита (Holding bit area)	H00000...H51115
Область вспомогательных битов	A00000...A95915
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область флагов задачи	TK0000...TK0031
Флаги состояния	ER, CY, N, OF, UF, >, =, <, >=, <>, <=, ON, OFF, AER
Тактовые импульсы	0.02 сек, 0.1 сек, 0.2 сек, 1 сек, 1 мин.
Область TR	—
Область DM	—
Область EM, не содержащая банков	—
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0 DR15, IR0 IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-)IR15

### Описание

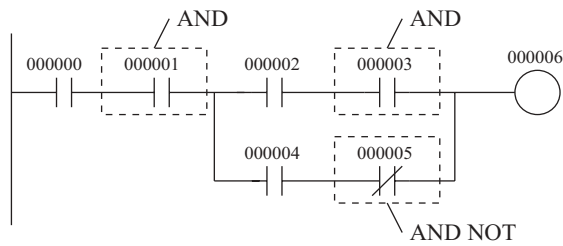
Команда AND NOT используется для нормально закрытого бита, подключенного последовательно. Команда AND NOT не может непосредственно подсоединиться к линии шины и не может использоваться в начале логического блока. Если немедленная регенерация не указывается, осуществляется чтение указанного бита в памяти ввода/вывода. Если немедленная регенерация указывается, осуществляется чтение состояния входной клеммы (ввода) Базового модуля ввода/вывода.

### Флаги

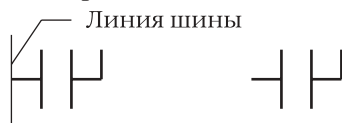
Не существует флагов, на которые эта команда оказывает влияние.

### Меры предосторожности

Для команды AND NOT может задаваться немедленная регенерация (!). Команда немедленной регенерации осуществляет обновление состояния бита ввода непосредственно перед выполнением команды для Базовых модулей ввода (однако, не для Базовых модулей ввода, находящихся в Slave панелях и не для Многоточечных модулей ввода C200H группы 2).

**Пример****3-3-5 OR: OR****Назначение**

Выполняет логическую операцию «ИЛИ» с состоянием ON/OFF бита указанного операнда и текущим условием выполнения.

**Символ релейно-контактной схемы****Модификации**

Модификации	Создает состояние ON в каждом цикле, когда результатом выполнения команды OR является состояние ON.	OR
	Создает состояние ON один раз при дифференцировании вверх	@OR
	Создает состояние ON один раз при дифференцировании вниз	%OR
Указание немедленной регенерации	!OR	
Комбинированные модификации	Осуществляет регенерацию бита ввода и создает состояние ON один раз при дифференцировании вверх.	!@OR
	Осуществляет регенерацию бита ввода и создает состояние ON один раз при дифференцировании вниз.	!%OR

**Применяемые области программы**

Область блоков	Область шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	OR бит-операнд
Область ввода/вывода (область CIO)	CIO 000000...CIO 614315
Рабочая область	W00000...W51115
Область удержания бита (Holding bit area)	H00000...H51115
Область вспомогательных битов	A00000...A95915
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область флагов задачи	TK0000...TK0031
Флаги состояния	ER, CY, N, OF, UF, >, =, <, >=, <=, ON, OFF, AER
Тактовые импульсы	0.02 сек, 0.1 сек, 0.2 сек, 1 сек, 1 мин.
Область TR	—
Область DM	—
Область EM, не содержащая банков	—
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—

Область	OR бит-операнд
Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0 DR15, IR0 IR15 ,IR0+(++)...IR15+(++) ,-(-)IR0...,-(-)IR15

**Описание**

Команда OR используется для нормально открытого бита, подключенного параллельно. Нормально открытый бит конфигурируется для выполнения логической операции ИЛИ с логическим блоком, начинающимся с команды LOAD или LOAD NOT (соединенной с линией шины или установленной в начале логического блока). Если немедленная регенерация не указывается, осуществляется чтение указанного бита в памяти ввода/вывода. Если немедленная регенерация указывается, осуществляется чтение состояния входной клеммы (ввода) Базового модуля ввода/вывода.

**Флаги**

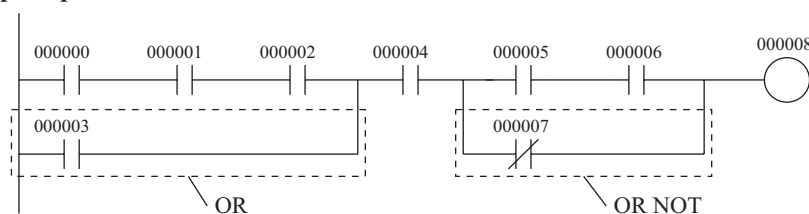
Не существует флагов, на которые эта команда оказывает влияние.

**Меры предосторожности**

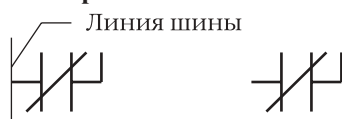
Для команды OR может задаваться дифференцирование вверх (@) или дифференцирование вниз (%). Если указывается дифференцирование вверх (@), условие выполнения переводится в состояние ON на время одного цикла только тогда, когда бит-операнд переходит из состояния OFF в состояние ON. Если указывается дифференцирование вниз (%), условие выполнения переводится в состояние ON на время одного цикла только тогда, когда бит-операнд переходит из состояния ON в состояние OFF.

Для команды OR может задаваться немедленная регенерация (!). Команда немедленной регенерации осуществляет обновление состояния бита ввода непосредственно перед выполнением команды для Базовых модулей ввода (однако, кроме Базовых модулей ввода, находящихся в Slave панелях и Многоточечных модулей ввода C200H группы 2).

Для команды OR можно объединять немедленную регенерацию и дифференцирование вверх или вниз (!@ или !%). При указании любой из этих команд ввод Базового модуля ввода подвергается регенерации непосредственно перед выполнением команды, и условие выполнения переводится в состояние ON на один цикл после перехода бита операнда из состояния OFF в состояние ON или наоборот.

**Пример****3-3-6 OR NOT: OR NOT****Назначение**

Выполняет логическую операцию «ИЛИ» с инвертированным состоянием указанного бита и текущим условием выполнения.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Создает состояние ON в каждом цикле, когда результатом выполнения команды OR NOT является состояние ON.	OR NOT
	Создает состояние ON один раз при дифференцировании вверх.	Не поддерживается.
	Создает состояние ON один раз при дифференцировании вниз.	Не поддерживается.
Указание немедленной регенерации	!OR NOT	
Комбинированные модификации	Осуществляет регенерацию бита ввода и создает состояние ON один раз при дифференцировании вверх.	Не поддерживается.
	Осуществляет регенерацию бита ввода и создает состояние ON один раз при дифференцировании вниз.	Не поддерживается.

**Применяемые области программы**

Область блоков	Область шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	OR NOT бит-операнд
Область ввода/вывода (область CIO)	CIO 000000...CIO 614315
Рабочая область	W00000...W51115
Область удержания бита (Holding bit area)	H00000...H51115
Область вспомогательных битов	A00000...A95915
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область флагов задачи	TK0000...TK0031
Флаги состояния	ER, CY, N, OF, UF, >, =, <, >=, <=, ON, OFF, AER
Тактовые импульсы	0.02 сек, 0.1 сек, 0.2 сек, 1 сек, 1 мин.
Область TR	—
Область DM	—
Область EM, не содержащая банков	—
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0 DR15, IR0 IR15 ,IR0+(++)...IR15+(++) ,(- -)IR0...(- -)IR15

**Описание**

Команда OR NOT используется для нормально закрытого бита, подключенного параллельно. Нормально закрытый бит конфигурируется для выполнения логической операции ИЛИ с логическим блоком, начинающимся с команды LOAD или LOAD NOT (соединенной с линией шины или установленной в начале логического блока). Если немедленная регенерация не указывается, осуществляется чтение указанного бита в памяти ввода/вывода. Если немедленная регенерация указывается, осуществляется чтение состояния входной клеммы (ввода) Базового модуля ввода/вывода.

**Флаги**

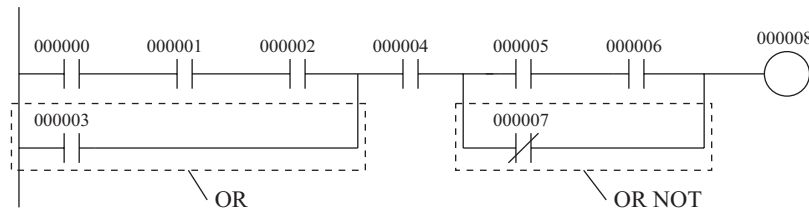
Не существует флагов, на которые эта команда оказывает влияние.

**Меры предосторожности**

Для команды OR может задаваться немедленная регенерация (!). Команда немедленной регенерации осуществляет обновление состояния бита ввода непосредственно перед выполнением команды для Базовых

модулей ввода (однако, кроме Базовых модулей ввода, находящихся в Slave панелях и Многоточечных модулей ввода C200H группы 2).

**Пример**

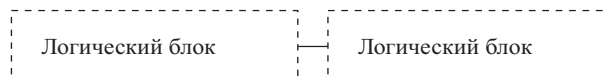


**3-3-7 AND LOAD: AND LD**

**Назначение**

Выполняет логическую операцию «И» с логическими блоками.

**Символ релейно-контактной схемы**



**Модификации**

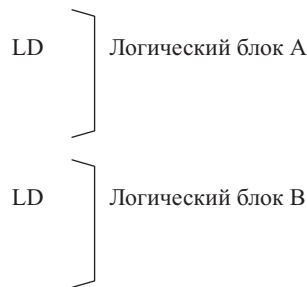
Модификации	Создает состояние ON в каждом цикле, когда результатом выполнения команды AND является состояние ON.	AND LD
Указание немедленной регенерации	Не поддерживается.	

**Применяемые области программы**

Область блоков	Область шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Описание**

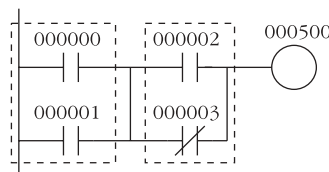
Команда AND LD соединяет последовательно один логический блок, находящийся непосредственно перед данной командой, с другим логическим блоком.



AND LD ---- Последовательное соединение логического блока А и логического блока В

Логический блок включает все команды от LOAD или LOAD NOT и заканчивается непосредственно перед следующей командой LOAD или LOAD NOT в этой же ступени.

На следующем ниже рисунке два логических блока обозначены прерывистыми линиями. Данный пример показывает, что состояние выполнения ON будет производиться тогда, когда состоянием выполнения левого блока является ON (т.е. CIO 000000 или CIO 000001 находится в состоянии ON), и когда состоянием выполнения правого блока также является ON (т.е. когда либо CIO 000002 находится в состоянии ON, либо CIO 000003 находится в состоянии OFF).

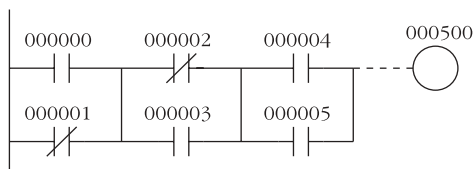


**Флаги**

Не существует флагов, на которые эта команда оказывает влияние.

**Меры предосторожности**

При помощи данной команды можно соединять последовательно три или более трех логических блоков. Вначале объедините два логических блока, затем подсоедините следующий блок. Допускается также располагать эту команду после трех или более логических блоков и затем соединять блоки последовательно. Когда логические блоки соединяются с помощью команд AND LOAD или OR LOAD, общее количество команд AND LOAD/OR LOAD должно соответствовать количеству команд LOAD/LOAD NOT минус 1. Если эти количество команд не соответствует указанному правилу, определяется ошибка команды.

**Пример****Пример программирования (1)**

Команда	Операнд
LD	000000
OR NOT	000001
LD NOT	000002
OR	000003
AND LD	—
LD	000004
OR	000005
AND LD	—
...	
OUT	000500

**Пример программирования (2)**

Команда	Операнд
LD	000000
OR NOT	000001
LD NOT	000002
OR	000003
LD	000004
OR	000005
...	
AND LD	—
AND LD	—
...	
OUT	000500

Команда AND LD может использоваться многократно. В способе программирования 2 (см. выше), тем не менее, количество команд AND LD на одну меньше, чем предшествующих команд LOAD и LOAD NOT.

В случае использования метода 2, убедитесь в том, что общее количество команд LOAD и LOAD NOT, введенных перед командой AND LOAD, не превышает восьми. Для использования девяти и более команд, применяйте метод программирования 1. Если в случае применения метода 2, используется девять и более команд LOAD и LOAD NOT, при проверке программы Периферийным устройством определяется ошибка программы.

**Программирование**

Адрес	Команда	Операнд
000000	LD	000000
000001	OR	000001
000002	LD	000002
000003	OR NOT	000003

Адрес	Команда	Операнд
000004	AND LD	—
000005	OUT	000500

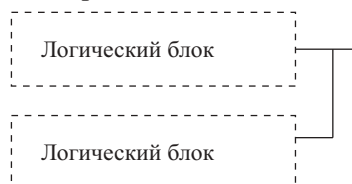
Вторая команда LD: используется для первого бита в следующем блоке, соединенном последовательно с предшествующим блоком.

### 3-3-8 OR LOAD: OR LD

#### Назначение

Выполняет логическую операцию «ИЛИ» с логическими блоками.

#### Символ релейно-контактной схемы



#### Модификации

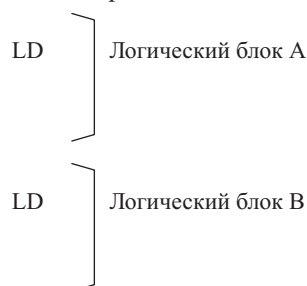
Модификации	Создает состояние ON в каждом цикле, когда результатом выполнения команды OR	OR LD
Указание немедленной регенерации	Не поддерживается.	

#### Areas Применяемые области программы

Область блоков	Область шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Описание

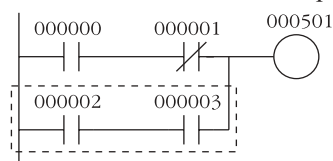
Команда OR LD осуществляет параллельное соединение логического блока, находящегося непосредственно перед данной командой, с другим логическим блоком.



OR LD ----- Параллельное соединение логического блока А и логического блока В

Логический блок включает все команды от LOAD или LOAD NOT и заканчивается непосредственно перед следующей командой LOAD или LOAD NOT в этой же ступени.

На следующем ниже рисунке между верхним и нижним логическими блоками необходимо ввести команду OR LOAD. Состояние выполнения ON будет производиться тогда, когда CIO 000000 находится в состоянии ON и CIO 000001 в состоянии OFF, или когда CIO 000002 и CIO 000003 находятся в состоянии ON. Выполнение действия и мнемоническое изображение команды для OR LOAD аналогично команде AND LOAD, за исключением того, что текущее состояние выполнения и последнее неиспользуемое состояние выполнения подвергаются операции OR.



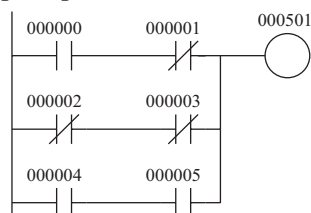
**Флаги**

Не существует флагов, на которые эта команда оказывает влияние.

**Меры предосторожности**

При помощи данной команды можно соединять параллельно три или более трех логических блоков. Вначале объедините два логических блока, затем подсоедините следующий блок. Допускается также располагать эту команду после трех или более логических блоков и затем соединять блоки параллельно.

Когда логические блоки соединяются с помощью команд AND LOAD или OR LOAD, общее количество команд AND LOAD/OR LOAD должно соответствовать количеству команд LOAD/LOAD NOT минус 1. Если это количество команд не соответствует указанному правилу, определяется ошибка команды.

**Пример****Пример программирования (1)**

Команда	Операнд
LD	000000
AND NOT	000001
LD NOT	000002
AND NOT	000003
OR LD	—
LD	000004
AND	000005
OR LD	—
...	
OUT	000501

**Пример программирования (2)**

Команда	Операнд
LD	000000
AND NOT	000001
LD NOT	000002
AND NOT	000003
LD	000004
AND	000005
OR LD	—
OR LD	—
...	
OUT	000501

Команда OR LD может использоваться неоднократно. В способе программирования 2 (см. выше), тем не менее, количество команд OR LD на одну меньше, чем предшествующих команд LOAD и LOAD NOT. В случае использования метода 2, убедитесь в том, что общее количество команд LOAD и LOAD NOT, введенных перед командой OR LOAD, не превышает восьми. Для использования девяти и более команд, применяйте метод программирования 1. Если в случае применения метода 2, используется девять и более команд LOAD и LOAD NOT, при проверке программы Периферийным устройством определяется ошибка программы.

**Программирование**

Адрес	Команда	Операнд
000000	LD	000000
000101	AND NOT	000001
000102	LD	000002

Адрес	Команда	Операнд
000103	AND	000003
000104	OR LD	–
000105	OUT	000501

Вторая команда LD: используется для первого бита в следующем блоке, соединенном последовательно с предшествующим блоком.

### 3-3-9 Дифференцированные команды и команды немедленной регенерации

В дополнение к обычным формам команды LOAD AND OR имеют модификации с дифференцированием и немедленной регенерацией. Кроме того, существует также две комбинации модифицированных команд. Кроме обычной формы команды LOAD NOT, AND NOT, OR NOT имеют модификации с немедленной регенерацией.

Время ввода/вывода данных при выполнении обычных команд отличается от времени обработки при выполнении дифференцированных команд, команд с немедленной регенерацией и дифференцированных команд с немедленной регенерацией.

Основные версии команд и модификации с дифференцированием выполняются с использованием данных ввода после предшествующей регенерации ввода/вывода, результаты выводятся при очередной операции ввода/вывода. В данном случае термин «регенерация ввода/вывода» означает обмен данными между встроенной памятью Модуля центрального процессора и Модулем ввода/вывода.

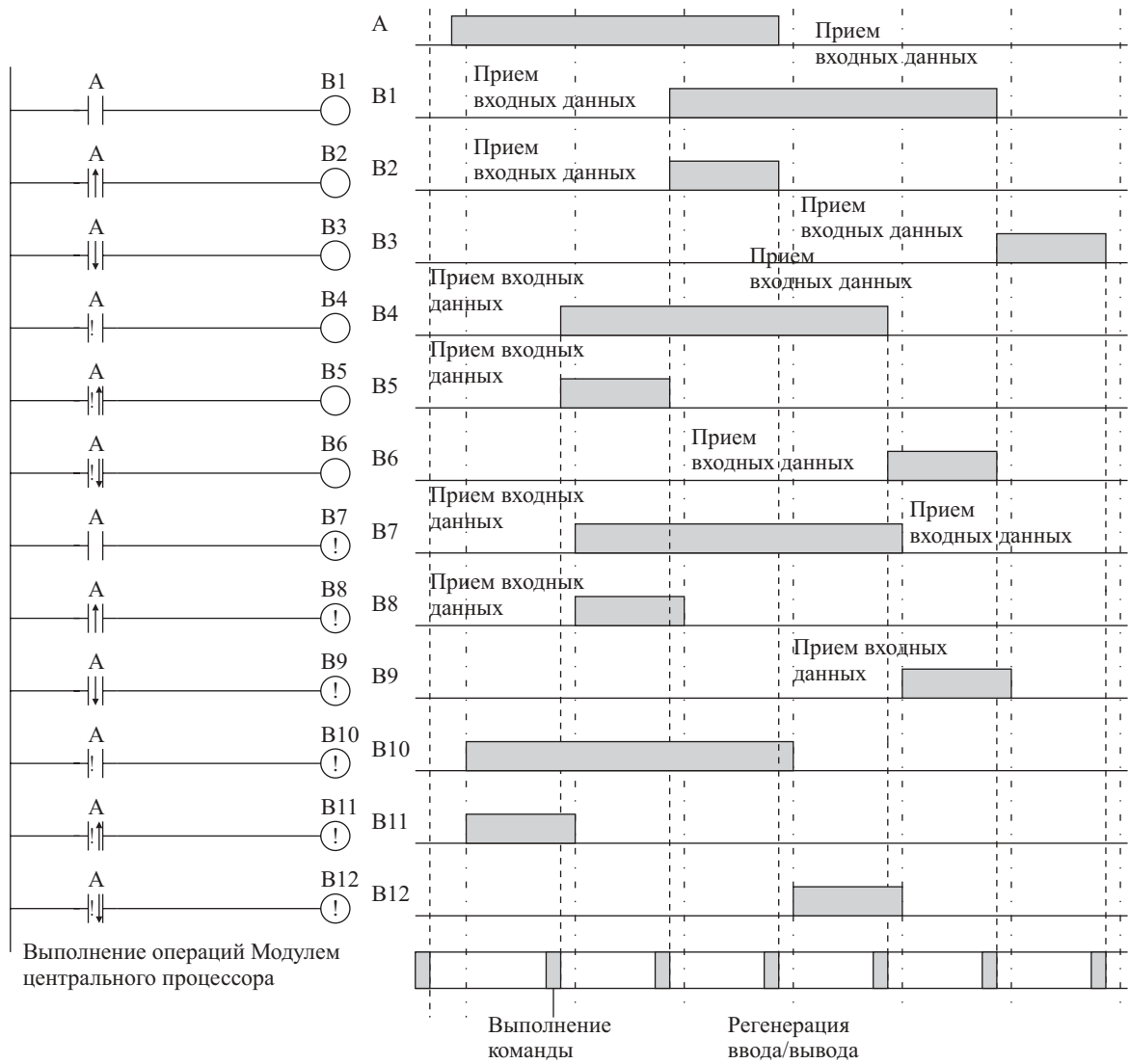
В дополнение к упомянутой выше регенерации ввода/вывода, команда немедленной регенерации осуществляет обмен данными с Модулем ввода/вывода для слов, которые доступны данной команде. Команда немедленной регенерации осуществляет регенерацию восьми битов последовательно (восемь битов младших разрядов или восемь битов старших разрядов) в дополнение к указанному биту.

Команда немедленной регенерации не может применяться к Модулям, находящимся в Slave-панелях.

Модификация команды	Мнемоническое изображение	Функция	Регенерация ввода/вывода
Обычная	LD, AND, OR, LD NOT, AND NOT, OR NOT	Состояние ON/OFF указанного бита берется Модулем центрального процессора при периодической регенерации и используется при следующем выполнении команды.	Периодическая регенерация
	OUT, OUT NOT	После выполнения команды, состояние ON/OFF указанного бита выводится при следующей периодической регенерации.	
Дифференцированные вверх	@LD, @ AND, @OR	Команда выполняется один раз, когда указанный бит переходит из состояния OFF в состояние ON, и это состояние удерживается в течение одного цикла.	
Дифференцированные вниз	%LD, %AND %OR	Команда выполняется один раз, когда указанный бит переходит из состояния ON в состояние OFF, и состояние ON удерживается в течение одного цикла.	
Немедленная регенерация	!LD, !AND, !OR, !LD NOT, !AND NOT, !OR NOT	Данные ввода для указанного бита берутся Модулем центрального процессора, и осуществляется выполнение команды.	Перед выполнением команды
	!OUT, !OUT NOT	После выполнения команды, осуществляется вывод данных указанного бита.	После выполнения команды
Дифференцированные вверх/немедленная регенерация	!@LD, !@AND, !@OR	Данные ввода для указанного бита регенерируются Модулем центрального процессора, и команда выполняется один раз, когда указанный бит переходит из состояния OFF в состояние ON. Состояние ON удерживается в течение одного цикла.	Перед выполнением команды
Дифференцированные вниз/немедленная регенерация	!%LD, !%AND, !%OR	Данные ввода для указанного бита регенерируются Модулем центрального процессора, и команда выполняется один раз, когда указанный бит переходит из состояния ON в состояние OFF. Состояние ON удерживается в течение одного цикла.	

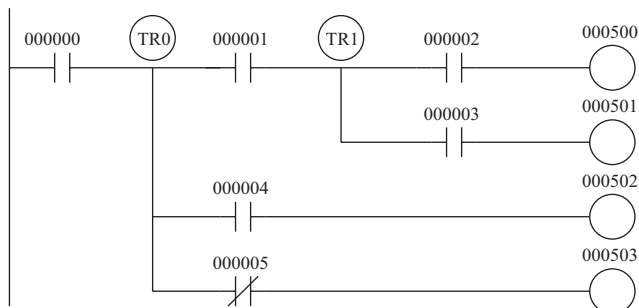
### 3-3-10 Временная диаграмма выполнения команд ввода/вывода

Следующие ниже графики показывают различие в распределении времени выполнения команд в программах, которые конфигурируются с применением команд LD и OUT.



### 3-3-11 Биты области TR

Биты области TR используются для временного сохранения состояния ON/OFF условия выполнения при составлении программы в мнемоническом коде. При программировании релейно-контактной программы область TR не используется, т.к. процесс выполняется Периферийным устройством автоматически. На следующем рисунке показан пример использования двух битов области TR.



Адрес	Команда	Операнды
000000	LD	000000
000001	OUT	TR 0
000002	AND	000001

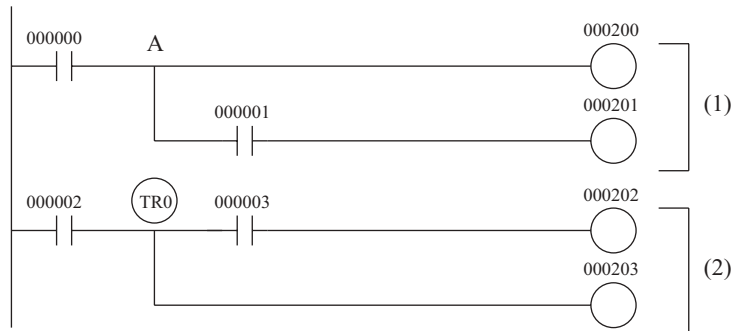


Адрес	Команда	Операнды
000003	OUT	TR 1
000004	AND	000002
000005	OUT	000500
000006	LD	TR 1
000007	AND	000003
000008	OUT	000501
000009	LD	TR 0
000010	AND	000004
000011	OUT	000502
000012	LD	TR 0
000013	AND NOT	000005
000014	OUT	000503

### Использование TR0...TR15

TR0...TR15 применяются только с командами LOAD и OUTPUT. Биты области TR могут применяться в любой последовательности, при этом на порядок использования битов никаких ограничений не налагается.

В некоторых случаях существует возможность упрощения программы, переписав ее таким образом, чтобы избежать применения битов области TR. На следующем ниже рисунке показан случай, когда использование битов области TR не является необходимостью, и случай, когда применение битов TR необходимо.



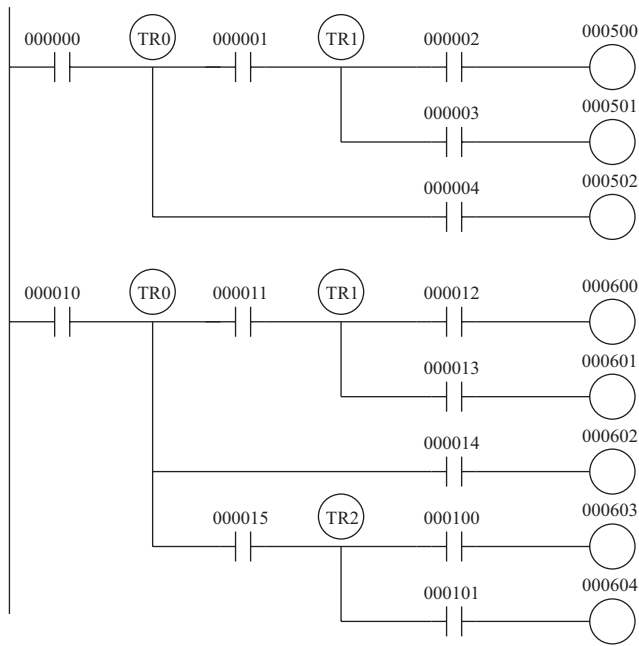
В блоке команд (1), состояние ON/OFF в точке А является таким же, как для вывода СЮ 00200, поэтому команда AND 000001 и команда OUT 000201 могут применяться без использования битов TR. В блоке команд (2) состояние в точке ответвления и состояние СЮ 000202 могут не соответствовать друг другу, поэтому применение битов TR необходимо. В данном случае количество шагов можно сократить, используя блок 1 вместо блока 2.

### Соображения по применению битов TR0...TR15

Биты области TR используются только для сохранения (OUT TR0 TR15) и восстановления (LD TR0...TR15) состояния ON/OFF точек разветвления в программах с многочисленными ветвями. Таким образом, эти биты отличаются от обычных битов, и не могут использоваться командами AND и OUT, или командами, содержащими NOT.

### Дублирование вывода TR0...TR15

Адрес бита области TR не может указываться дважды в одном блоке команд программы, содержащей много ветвей, как показано на следующем ниже рисунке. Тем не менее, этот адрес может использоваться в другом блоке.

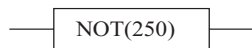


### 3-3-12 NOT: NOT(520)

**Назначение**

Осуществляет инвертирование состояния выполнения.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Инвертирует состояние выполнение в каждом из циклов	NOT(520)
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Область блоков	Область шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Описание**

Команда NOT(520) располагается между состоянием выполнения и следующей командой для инвертирования условия выполнения.

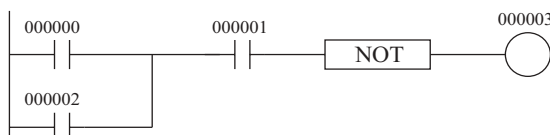
**Флаги**

Не существует флагов, на которые эта команда оказывает влияние.

**Меры предосторожности**

Команда NOT(520) является промежуточной командой, т.е. она не может использоваться в качестве последней команды. Непременно программируйте следующую команду после команды NOT(512).

**Пример**



Следующая ниже таблица показывает работу этой части программы.

Состояние входного бита			Состояние выходного бита
СЮ 000000	СЮ 000001	СЮ 000002	СЮ 000003
1	1	1	0

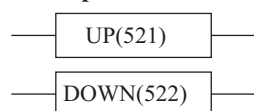
Состояние входного бита			Состояние выходного бита	
1	1	0	0	0
1	0	1	1	1
0	1	1	0	0
1	0	0	0	1
0	1	0	0	1
0	0	1	1	1
0	0	0	0	1

### 3-3-13 Условие ON/OFF : UP(521) и DOWN(522)

#### Назначение

Команда UP(521) осуществляет перевод условия выполнения следующей команды в состояние ON на время одного цикла, когда принимаемое командой (UP(521)) условие выполнения переходит из состояния OFF в состояние ON. Команда DOWN(522) осуществляет перевод условия выполнения для следующей команды в состояние ON на время одного цикла, когда принимаемое командой (DOWN(522)) условие выполнения переходит из состояния ON в состояние OFF.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Создает состояние ON один раз для дифференцирования вверх	UP(521)
Указание немедленной регенерации	Не поддерживается	
Модификации	Создает состояние ON один раз для дифференцирования вниз.	DOWN(522)
Указание немедленной регенерации.	Не поддерживается	

#### Применяемые области программы

Область блоков	Область шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Описание

Команда UP(521) располагается между условием выполнения и следующей командой для перевода условия выполнения в состояние, обусловленное дифференцированием вверх. Команда UP(521) вызывает выполнение присоединенной команды один раз, когда условие выполнения переходит из состояния OFF в состояние ON.

Команда DOWN(522) располагается между условием выполнения и следующей командой для перевода условия выполнения в состояние, обусловленное дифференцированием вниз. Команда DOWN(522) вызывает выполнение присоединенной команды один раз, когда условие выполнения переходит из состояния ON в состояние OFF.

Команды DIFU(013) и DIFU(014) могут применяться для аналогичных целей, однако они требуют использования рабочих битов. Использование команд UP(521) и DOWN(522) упрощают процесс программирования, уменьшая требуемое количество рабочих битов и адресов программы.

#### Флаги

Команды UP(521) и DOWN(522) не влияют на состояние флагов.

#### Меры предосторожности

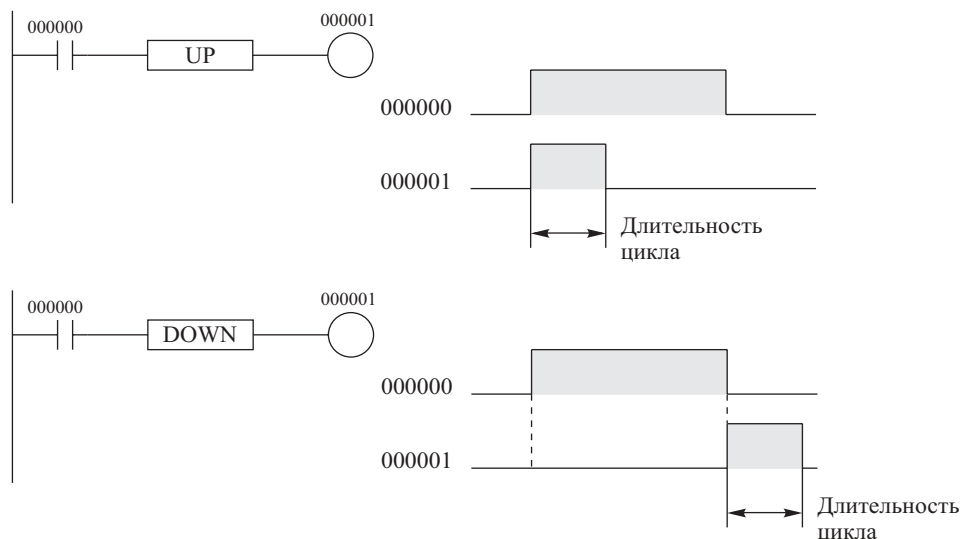
Команды UP(521) и DOWN(522) являются промежуточными командами, т.е. они не могут использоваться в качестве последних команд. Непременно программируйте следующую команду после команд UP(521) и DOWN(522).

Функционирование команд UP(521) и DOWN(522) зависит от условия выполнения команды так же, как от условия выполнения раздела программы, когда команда находится в заблокированной части программы, обходимой части программы, или подпрограммы. Для ознакомления с подробностями обратитесь к разде-

лам 3-5-5 «*INTERLOCK и INTERLOCK CLEAR: IL(002) и ILC(003)*», «*JUMP и JUMP END: JMP(004) и JME(005)*» а также к разделу 3-19 «*Команды управления прерываниями*».

### Примеры

В следующем примере, когда CIO 000000 переходит из состояния OFF в состояние ON, CIO 000001 переводится в состояние ON на время одного цикла.



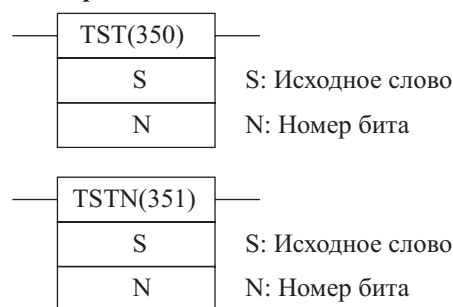
### 3-3-14 Команда BIT TEST: TST(350) и TSTN(351)

#### Назначение

Команды LD TST(350), AND TST(350) и OR TST(350) используются в программах подобно командам LD, AND и OR. Условие выполнения находится в состоянии ON, когда указанный бит в заданном слове находится в состоянии ON и, соответственно, условие выполнения находится в состоянии OFF, когда указанный бит заданного слова находится в состоянии OFF.

Команды LD TSTN(351), AND TSTN(351) и OR TSTN(351) используются в программах подобно командам LD NOT, AND NOT и OR NOT. Условие выполнения находится в состоянии OFF, когда указанный бит в заданном слове находится в состоянии ON и наоборот, условие выполнения в состоянии ON, когда указанный бит заданного слова находится в состоянии OFF.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле	TST(350)
Указание немедленной регенерации	Не поддерживается	
Модификации	Выполняется в каждом цикле	TSTN(351),
Указание немедленной регенерации.	Не поддерживается	

#### Применяемые области программы

Область блоков	Область шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****N: номер бита**

Номер бита должен находиться между 0000 и 000F в шестнадцатеричном коде, или между &0000 и &0015 в десятичном коде. При указании адреса слова действителен только младший бит слова (0...F в шестнадцатеричном коде).

**Спецификации операндов**

Область	S	N
Область ввода/вывода (область CIO)	CIO 000000...CIO 614315	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A000...A959	
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_32767 (n=0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000... *En_32767 (n=0...C)	
Константы	–	#0000 - #000F (двоичные) или &0 - &15
Регистры данных	DR0...DR15	
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)... ,IR15+(++) ,-( - )IR0...,-( - )IR15	

**Описание**

Команды LD TST(350), AND TST(350) и OR TST(350) используются в программах подобно командам LD, AND и OR. Условие выполнения находится в состоянии ON, когда указанный бит в заданном слове находится в состоянии ON и соответственно, условие выполнения в состоянии OFF, когда указанный бит заданного слова находится в состоянии OFF.

Команды LD TSTN(351), AND TSTN(351) и OR TSTN(351) используются в программах подобно командам LD NOT, AND NOT и OR NOT. Условие выполнения находится в состоянии OFF, когда указанный бит в заданном слове находится в состоянии ON и наоборот, условие выполнения в состоянии ON, когда указанный бит заданного слова находится в состоянии OFF.

**Флаги**

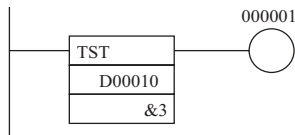
Наименование	Знак	Работа
Флаг ошибки	ER	OFF
Флаг равенства	=	OFF
Флаг отрицательного значения	N	OFF

**Меры предосторожности**

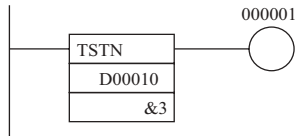
Команды TST(350) и TSTN(351) являются промежуточными командами, т.е. они не могут использоваться в качестве последних команд. Непременно программируйте следующую команду после команд TST(350) и TSTN(351).

**Пример****LD TST(350) и LD TSTN(351)**

В следующем ниже примере CIO 000001 переводится в состояние ON, когда бит 3 в D00010 находится в состоянии ON.

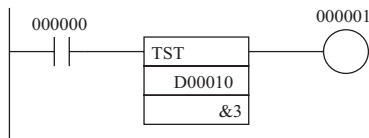


В следующем ниже примере CIO 000001 переводится в состояние ON, когда бит 3 в D00010 находится в состоянии OFF.

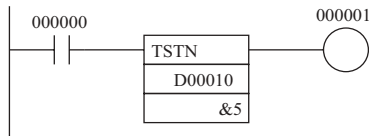


#### AND TST(350) и AND TSTN(351)

В следующем ниже примере CIO 000001 переводится в состояние ON, когда CIO 000000 и бит 3 в D00010 находятся в состоянии ON.

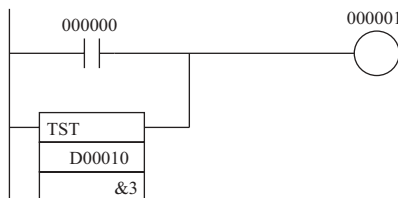


В следующем ниже примере CIO 000001 переводится в состояние ON, когда CIO 000000 находится в состоянии ON, а бит 3 в D00010 находится в состоянии OFF.

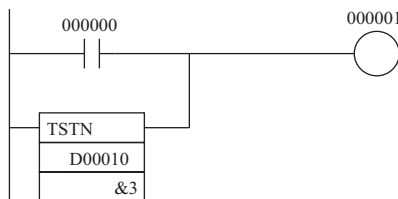


#### OR TST(350) и OR TSTN(351)

В следующем ниже примере CIO 000001 переводится в состояние ON, когда CIO 000000 или бит 3 в D00010 находятся в состоянии ON.



В следующем ниже примере CIO 000001 переводится в состояние ON, когда CIO 000000 находится в состоянии ON, а бит 3 в D00010 находится в состоянии OFF.

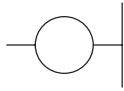


## 3-4 Команды последовательного вывода

## 3-4-1 OUTPUT: OUT

**Назначение**

Осуществляет вывод результата (состояния выполнения) выполнения логической операции в заданный бит.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	OUT
	Выполняется один раз для дифференцирования вверх	Не поддерживается.
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		!OUT

**Применяемые области программы**

Область блоков	Область шагов	Подпрограммы	Задачи прерывания
Не применяется	Да	Да	Да

**Спецификации операндов**

Область	OUT бит-операнд
Область ввода/вывода (область CIO)	CIO 000000...CIO 614315
Рабочая область	W00000...W51115
Область удержания бита (Holding bit area)	H000000...H51115
Область вспомогательных битов	A44800...A95915
Область таймера	—
Область счетчика	—
Область TR	TR0–TR15
Область DM	—
Область EM, не содержащая банков	—
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0 ,(- -)IR15

**Описание**

Если немедленная регенерация не указывается, осуществляется запись состояния условия выполнения (присутствие напряжения) в указанный бит в памяти ввода/вывода. Если немедленная регенерация указывается, запись состояния условия выполнения (присутствие напряжения) производится не только в выходной бит памяти ввода/вывода, но и на выход Базового модуля вывода.

**Флаги**

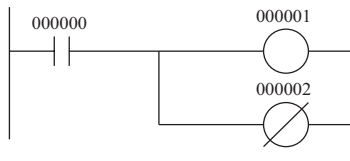
Не существует флагов, на которые эта команда оказывает влияние.

**Меры предосторожности**

Для команд OUT и OUT NOT может задаваться немедленная регенерация (!). Команда немедленной регенерации осуществляет обновление состояния вывода непосредственно после выполнения команды для Базовых модулей вывода (однако, кроме Базовых модулей вывода, находящихся в Slave панелях и Много-



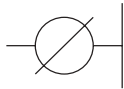
точечных модулей вывода С200Н группы 2). Кроме того, при выполнении этой команды осуществляется запись условия выполнения (присутствие питания) в заданный бит памяти ввода/вывода.

**Пример**

Команда	Операнд
LD	000000
OUT	000001
OUT NOT	000002

**3-4-2 OUTPUT NOT: OUT NOT****Назначение**

Осуществляет инвертирование и вывод результата (состояния выполнения) выполнения логической операции в заданный бит.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	OUT NOT
	Выполняется один раз для дифференцирования вверх	Не поддерживается.
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		!OUT NOT

**Применяемые области программы**

Область блоков	Область шагов	Подпрограммы	Задачи прерывания
Не применяется	Да	Да	Да

**Спецификации операндов**

Область	OUT бит-операнд
Область ввода/вывода (область CIO)	CIO 000000...CIO 614315
Рабочая область	W00000...W51115
Область удержания бита (Holding bit area)	H000000...H51115
Область вспомогательных битов	A44800...A95915
Область таймера	—
Область счетчика	—
Область TR	TR0...TR15
Область DM	—
Область EM, не содержащая банков	—
гi-40Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	—
Регистры данных	—
Индексные регистры	—

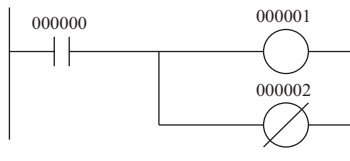
Область	OUT бит-операнд
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15

**Описание**

Если немедленная регенерация не указывается, осуществляется инвертирование и запись состояния условия выполнения (присутствие напряжения) в указанный бит в памяти ввода/вывода. Если немедленная регенерация указывается, осуществляется инвертирование и запись состояния условия выполнения (присутствие напряжения) не только в выходной бит памяти ввода/вывода, но и на выход Базового модуля.

**Флаги**

Не существует флагов, на которые эта команда оказывает влияние.

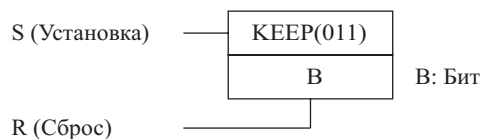
**Пример**

Команда	Операнд
LD	000000
OUT	000001
OUT NOT	000002

## 3-4-3 KEEP: KEEP(011)

**Назначение**

Функционирует подобно реле с фиксацией.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	KEEP(011)
	Выполняется один раз для дифференцирования вверх.	Не поддерживается.
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		! KEEP(011)

**Применяемые области программы**

Область блоков	Область шагов	Подпрограммы	Задачи прерывания
Не применяется	Да	Да	Да

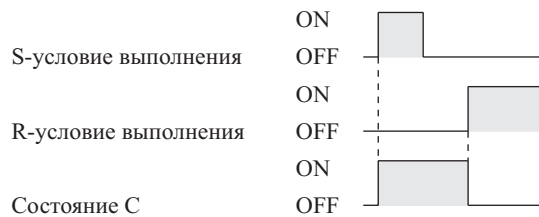
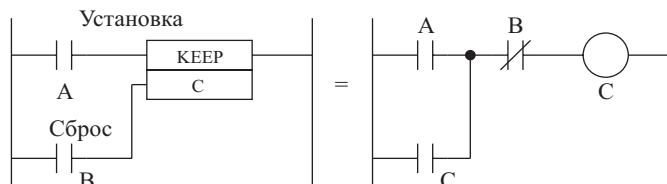
**Спецификации операндов**

Область	LD бит-операнд
Область ввода/вывода (область CIO)	CIO 000000...CIO 614315
Рабочая область	W00000...W51115
Область удержания бита (Holding bit area)	H000000...H51115
Область вспомогательных битов	A44800...A95915
Область таймера	-
Область счетчика	-
Область DM	-
Область EM, не содержащая банков	-

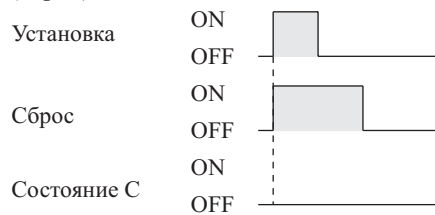
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...IR15(++) ,(- -)IR0...(- -)IR15

### Описание

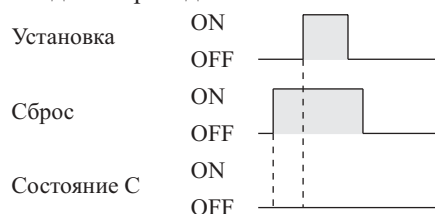
Когда S переходит в состояние ON, бит назначения переводится в состояние ON и остается в этом состоянии до сброса, вне зависимости от того, остается ли S в состоянии ON или переходит в состояние OFF. Когда R переводится в состояние ON, бит назначения переводится в состояние OFF. Соотношение между условиями выполнения и состоянием бита при выполнении команды KEEP(011) приводится на следующем рисунке.



Когда S и R одновременно переводятся в состояние ON, преимущественное значение имеет состояние R (сброс).

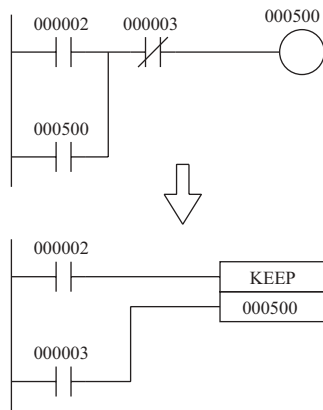


Когда R переводится в состояние ON, сигнал, подаваемый на вход S (установка), не читается.

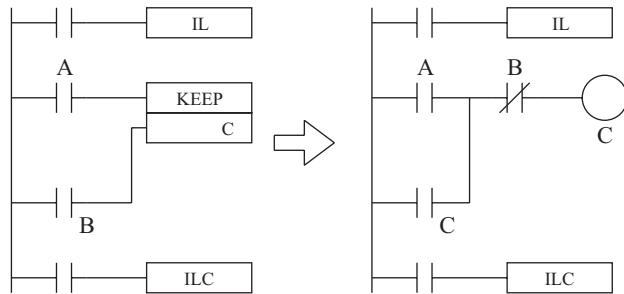


Команда KEEP(011) имеет модификацию с немедленной регенерацией !KEEP(011). Когда в команде KEEP(011) в качестве операнда «B» задается внешний выходной бит, любое изменение в «B» подвергается регенерации при выполнении команды KEEP(011) и немедленно отражается на состоянии выходного бита. (Изменения не отражаются на состоянии битов, распределенных Модулям группы 2, Специальным модулям ввода/вывода, или Модулям, установленным в Slave-панелях SYSMAC BUS.)

Команда KEEP(011) функционирует подобно биту, самостоятельно поддерживающему свое состояние, однако применение этой команды позволяет использовать на одну команду меньше.



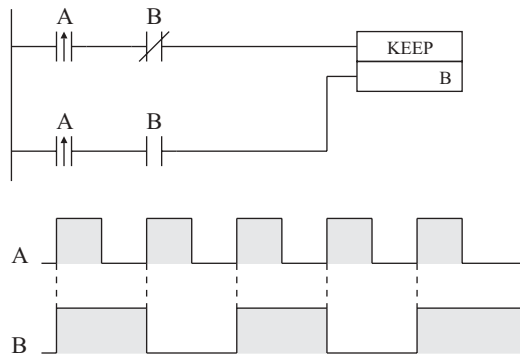
Биты, запрограммированные с помощью команды KEEP(011), удерживают свое состояние даже в блокированных секциях программы, в отличие от битов, самостоятельно удерживающих состояние без помощи команды KEEP(011).



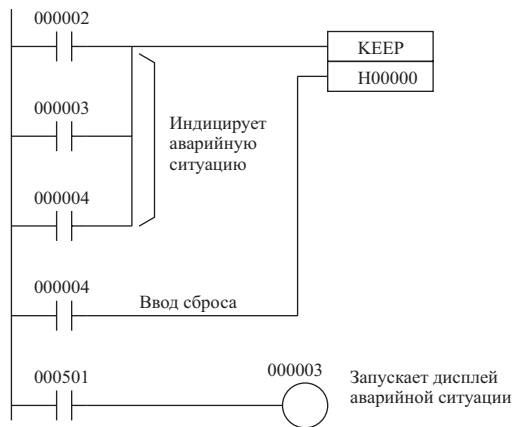
Выходной бит удерживает предшествующее состояние в блокированной секции программы

Выходной бит в блокированной секции программы будет переведен в состояние OFF

Команда KEEP(011) может использоваться для создания триггерной схемы, как показано на следующем рисунке.



Если удерживающийся бит используется для «В», состояние бита будет сохраняться даже в случае прерывания подачи питания. Таким образом, команда KEEP(011) может применяться для программирования битов, которые будут поддерживать состояние при перезапуске Программируемого контроллера после прерывания подачи питания. Пример, который может использоваться для создания аварийного дисплея, появляющегося вслед за возникновением аварийной ситуации и выключением системы, показан на следующем рисунке.



Состояние битов области ввода/вывода может сохраняться в случае прерывания подачи питания посредством перевода бита IOM Hold Bit в состояние ON и заданием удержания IOM Hold Bit в начальных установках Программируемого контроллера. В этом случае биты области ввода/вывода, используемые в команде KEEP(011), будут поддерживать состояние после перезапуска Программируемого контроллера вслед за прерыванием подачи питания, подобно удерживающим битам. После изменения начальных установок непременно осуществите перезапуск Программируемого контроллера, в противном случае изменения не будут введены в начальные установки.

#### Флаги

Не существует флагов, на которые команда KEEP(011) оказывает влияние.

#### Меры предосторожности

В случае, когда входное устройство использует источник питания переменного тока, никогда не применяйте входной бит в нормально-замкнутом состоянии в качестве сигнала переустановки (R) для команды KEEP(011). Задержка при выключении источника питания постоянного тока Программируемого контроллера (относится к источнику питания переменного тока для входного устройства) может привести к переустановке бита команды KEEP(011). Подобная ситуация представлена на следующем ниже рисунке.

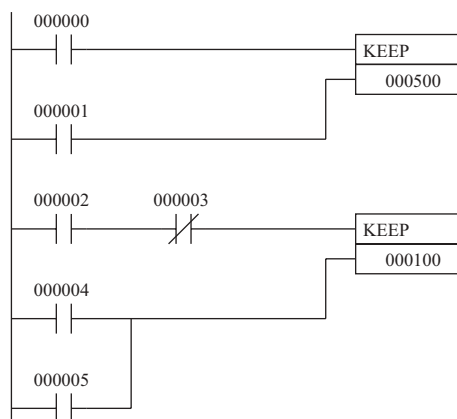


Порядок ввода операндов для команды KEEP(011) различен для ввода в релейно-контактную схему и для ввода в мнемоническом виде.

- Порядок ввода в релейно-контактную схему: Ввод установки → KEEP(011) → Ввод сброса.
- Порядок ввода в мнемоническом коде: Ввод установки → Ввод сброса → KEEP(011).

#### Пример

В следующем примере, когда CIO 000000 переходит в состояние ON, CIO 00500 переводится в состояние ON. CIO 00500 остается в состоянии ON до тех пор, пока CIO 000001 не перейдет в состояние ON. В следующем примере, когда CIO 000002 переходит в состояние ON и CIO 000003 переходит в состояние OFF, CIO 00100 переводится в состояние ON. Это состояние поддерживается до тех пор, пока CIO 000004 или CIO 000005 не перейдет в состояние ON.



### Составление программы

Адрес	Команда	Операнд
000100	LD	000000
000101	LD	000001
000102	KEEP(011)	000500
000103	LD	000002
000104	AND NOT	000003
000105	LD	000004
000106	OR	000005
000106	KEEP(011)	000100

**Примечание:** Порядок ввода команды *KEEP(011)* различен для ввода в релейно-контактную схему и для ввода в мнемоническом виде. При вводе в релейно-контактную схему задайте ввод установки, введите *KEEP(011)*, затем задайте ввод переустановки. При вводе в мнемонической форме задайте ввод установки, ввод переустановки затем введите *KEEP(011)*.

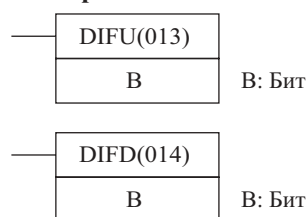
### 3-4-4 Команда дифференцирования вверх/вниз DIFFERENTIATE UP/DOWN: DIFU(013) и DIFD(014)

#### Назначение

По команде DIFU(013) осуществляется перевод бита назначения в состояние ON на один цикл, когда условие выполнения переходит из состояния OFF в состояние ON (по восходящему фронту).

По команде DIFD(014) осуществляется перевод бита назначения в состояние ON на один цикл, когда условие выполнения переходит из состояния ON в состояние OFF (по ниспадающему фронту).

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	Не поддерживается.
	Выполняется один раз для дифференцирования вверх	DIFU(013).
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		!DIFU(013).
Модификации	Выполняется в каждом цикле для состояния ON.	Не поддерживается.
	Выполняется один раз для дифференцирования вверх	DIFD(014).
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.

Указание немедленной регенерации.	!DIFD(014)
-----------------------------------	------------

**Применяемые области программы**

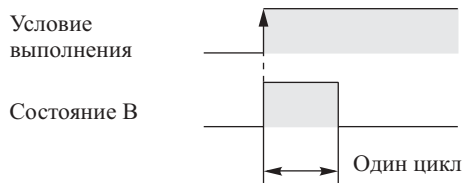
Область блоков	Область шагов	Подпрограммы	Задачи прерывания
Не применяется	Да	Да	Да

**Спецификации операндов**

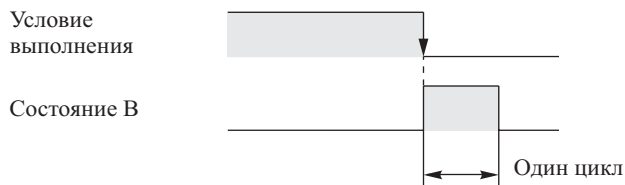
Область	В
Область ввода/вывода (область CIO)	CIO 000000...CIO 614315
Рабочая область	W00000...W51115
Область удержания бита (Holding bit area)	H000000...H51115
Область вспомогательных битов	A44800...A95915
Область таймера	—
Область счетчика	—
Область DM	—
Область EM, не содержащая банков	—
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,-(-)IR0...,-(-)IR15

**Описание**

Когда условие выполнения переходит из состояния OFF в состояние ON, команда DIFU(013) переводит «В» в состояние ON. Когда в следующем цикле встречается следующая команда DIDU(013), «В» переводится в состояние OFF.



Когда условие выполнения переходит из состояния ON в состояние OFF, команда DIFD(014) переводит «В» в состояние ON. Когда в следующем цикле встречается следующая команда DIDU(014), «В» переводится в состояние OFF.



Команды DIFU(013) и DIFD(014) имеют модификации с немедленной регенерацией (!DIFU(013) и !DIFD(014)). Когда в одной из этих команд бит внешнего вывода указан в качестве операнда В, любое изменение В будет подвержено регенерации после выполнения команды, после чего немедленно отразится на состоянии бита вывода. (Изменения не отражаются на состоянии битов, распределенных Модулям группы 2, Специальным модулям ввода/вывода, или Модулям, установленным в Slave-панелях SYSMAC BUS.)

Команды UP(521) и DOWN(522) могут использоваться для выполнения команды только в одном цикле, когда условие выполнения переходит из состояния OFF в состояние ON или наоборот. Для детального ознакомления обратитесь к разделу 3-3-13 **CONDITION ON/OFF: UP(521) и DOWN(522)**.



**Флаги**

Не существует флагов, на которые оказывают влияние команды DIFU(013) и DIFD(014).

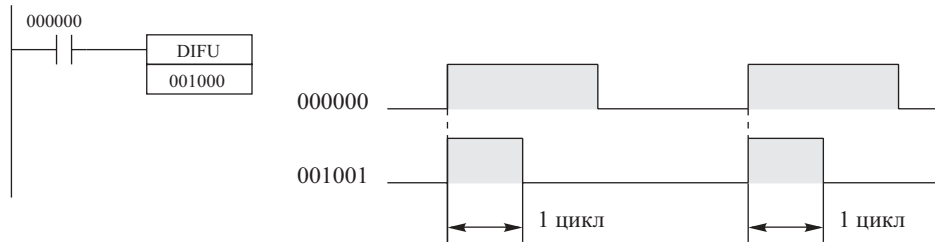
**Меры предосторожности**

Функционирование команд DIFU(013) и DIFD(014) зависит от условия выполнения команды так же, как от условия выполнения для раздела программы, когда команда находится в блокированной части программы, обходимой части программы, или подпрограммы. Для ознакомления с подробностями обратитесь к разделам 3-5-5 «*INTERLOCK и INTERLOCK CLEAR: IL(002) и ILC(003)*», «*JUMP и JUMP END: JMP(004) и JME(005)*» а также к разделу 3-19 «*Команды управления прерываниями*».

**Примеры**

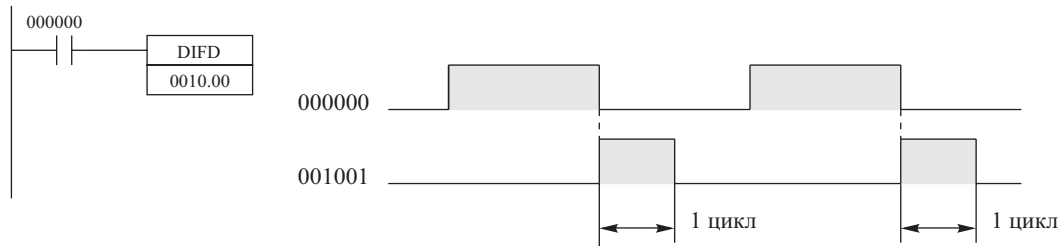
**Функционирование команды DIFU(013)**

В следующем примере, когда CIO 000000 переходит из состояния OFF в состояние ON, CIO 000100 переводится в состояние ON на время одного цикла.



**Функционирование команды DIFD(014)**

В следующем примере, когда CIO 000000 переходит из состояния ON в состояние OFF, CIO 000100 переводится в состояние ON на время одного цикла.



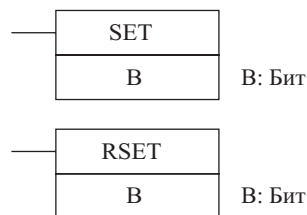
3-4-5 Команды SET и RESET: SET и RSET

**Назначение**

По команде SET осуществляется перевод бита операнда в состояние ON, когда условие выполнения находится в состоянии ON.

По команде RSET осуществляется перевод бита операнда в состояние OFF, когда условие выполнения находится в состоянии ON.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	SET
	Выполняется один раз для дифференцирования вверх.	@SET
	Выполняется один раз для дифференцирования вниз.	%SET.
Указание немедленной регенерации.		!SET
Комбинированные модификации	Немедленная регенерация выполняется один раз для дифференцирования вверх.	!@SET
	Немедленная регенерация выполняется один раз для дифференцирования вниз.	!%SET.

Модификации	Выполняется в каждом цикле для состояния ON.	RSET
	Выполняется один раз для дифференцирования вверх	@RSET
	Выполняется один раз для дифференцирования вниз.	%RSET.
Указание немедленной регенерации.		!RSET
Комбинированные модификации	Немедленная регенерация выполняется один раз для дифференцирования вверх.	!@RSET
	Немедленная регенерация выполняется один раз для дифференцирования вниз.	!%RSET.

### Применяемые области программы

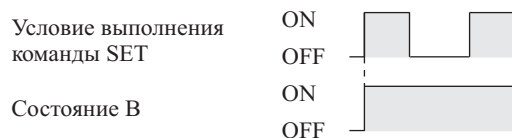
Область блоков программы	Область шагов программы	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

### Спецификации операндов

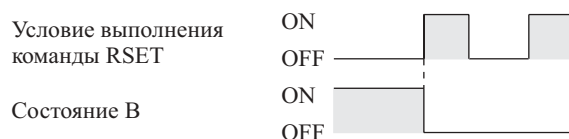
Область	B
Область ввода/вывода (область CIO)	CIO 000000...CIO 614315
Рабочая область	W00000...W51115
Область удержания бита (Holding bit area)	H000000...H51115
Область вспомогательных битов	A44800...A95915
Область таймера	—
Область счетчика	—
Область DM	—
Область EM, не содержащая банков	—
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15

### Описание

По команде SET производится перевод бита операнда в состояние ON, когда условие выполнения находится в состоянии ON. Команда не отражается на состоянии бита, когда условие выполнения находится в состоянии OFF. Для перевода бита, установленного в состояние ON командой SET, выполните команду RSET.



По команде RSET производится перевод бита операнда в состояние OFF, когда условие выполнения находится в состоянии ON. Команда не отражается на состоянии бита, когда условие выполнения находится в состоянии OFF. Для перевода бита, установленного в состояние OFF командой RSET, выполните команду SET.



Команды SET и RSET имеют модификации с немедленной регенерацией (!SET и !RSET). Когда в одной из этих команд бит внешнего вывода указан в качестве операнда B, любое изменение B будет подвержено регенерации после выполнения команды, после чего немедленно отразится на состоянии бита вывода. (Из-

менения не отражаются на состоянии битов, распределенных Модулям группы 2, Специальным модулям ввода/вывода, или Модулям, установленным в Slave-панелях SYSMAC BUS.)

Вводы установки и сброса для команды KEEP(011) должны программироваться при помощи команды, а команды SET и RSET могут программироваться абсолютно независимо. Более того, один и тот же бит может использоваться в качестве операнда в любом количестве команд SET и RSET.

#### Флаги

Не существует флагов, на которые оказывают влияние команды SET и RSET.

#### Меры предосторожности

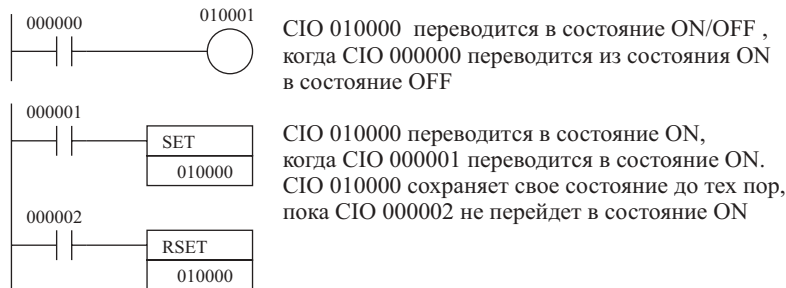
Команды SET и RSET не могут применяться для установки и сброса таймеров и счетчиков.

Когда команды SET и RSET располагаются между командами IL(002) и ILC(003) или JMP(004) и JMPE(005), состояние указанного бита не будет изменяться, если данный раздел программы заблокирован или обходится.

#### Пример

Различия между командами OUT/OUT NOT и SET/RSET

Функционирование команды SET отличается от команды OUT, т.к. по команде OUT осуществляется перевод бита операнда в состояние OFF, когда условие выполнения находится в состоянии OFF. Подобно изложенному выше, команда RSET отличается от команды OUT NOT, т.к. по команде OUT NOT осуществляется перевод бита операнда в состояние ON, когда условие выполнения находится в состоянии OFF.



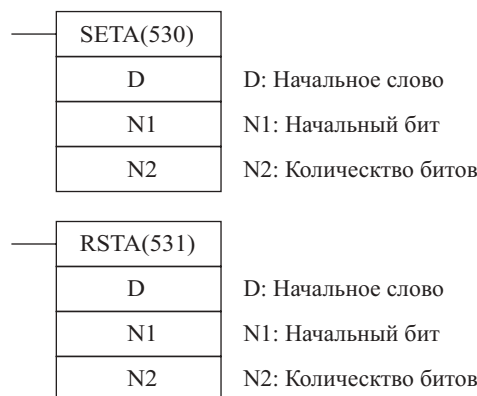
### 3-4-6 Команда установки и сброса множества битов MULTIPLE BIT SET/RESET: SETA(530)/RSTA(531)

#### Назначение

По команде SETA(530) осуществляется перевод заданного количества битов в состояние ON.

По команде RSTA(531) осуществляется перевод заданного количества битов в состояние OFF.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	SETA(530)
	Выполняется один раз для дифференцирования вверх	@ SETA(530)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается.

Модификации	Выполняется в каждом цикле для состояния ON.	RSTA(531)
	Выполняется один раз для дифференцирования вверх	@RSTA(532)

	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается.

**Применяемые области программы**

Область блоков	Область шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****D: начальное слово**

Указывает первое слово, в котором биты будут изменять свое состояние (ON/OFF).

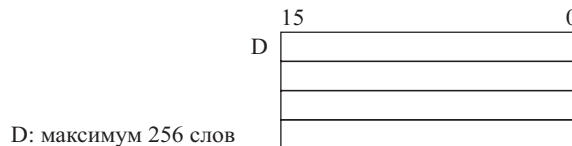
**N1: Начальный бит**

Указывает первый бит, который будет изменять свое состояние (ON/OFF). Число N1 должно находиться в следующих пределах: #0000 - #000F (&0 - &15).

**N2: количество битов**

Указывает количество битов, которые будут переводиться в состояние ON или OFF. Число N2 должно находиться в следующих пределах: #0000 - #FFFF (&0 - &65535).

**Примечание:** Биты, состояние которых подлежит изменению, должны находиться в одной и той же области данных. (Диапазон слов примерно равен  $D + N2 + 16$ .)

**Спецификации операндов**

Область	D	N1	N2
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A448...A959	A000...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n=0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0 C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0C)		
Константы	-	#0000...#000F (двоичные) или &0 &15	#0000...#FFFF (двоичные) или &0 &65535
Регистры данных	DR0...DR15		
Индексные регистры	...		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-(-)IR15		

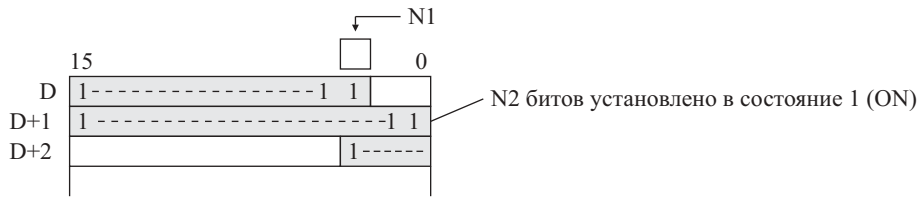
**Описание**

Функционирование команд SETA(530) и RSTA(531) приводится ниже, отдельно для каждой из команд.

**Функционирование команды SETA(530)**

Команда SETA(530) осуществляет перевод N2 битов в состояние ON, начиная с бита N1, и кончая старшими битами слова D. Состояние всех прочих битов остается неизменным. (Изменения не производятся, если N2 установлено в значение, равное нулю.)

Биты, переведенные в значение ON при помощи команды SETA(530), могут переводиться в состояние OFF как при помощи команды RSTA(531), так и при помощи любой другой команды.

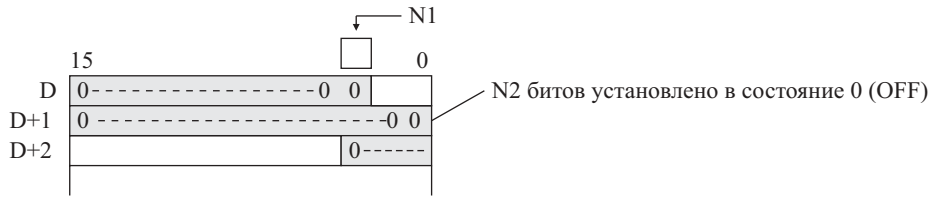


Команда SETA(530) может применяться для перевода в состояние ON битов областей данных, доступ к которым осуществляется только указанием слов, например областей DM и EM.

**Функционирование команды RSTA(531)**

Команда RSTA(531) осуществляет перевод N2 битов в состояние OFF, начиная с бита N1, и кончая старшими битами слова D. Состояние всех прочих битов остается неизменным. (Изменения не производятся, если N2 установлено в значение, равное нулю.)

Биты, переведенные в значение OFF при помощи команды RSTA(531), могут переводиться в состояние ON как при помощи команды SETA(530), так и при помощи любой другой команды.



Команда RSTA(531) может применяться для перевода в состояние OFF битов областей данных, доступ к которым осуществляется только указанием слов, например областей DM и EM.

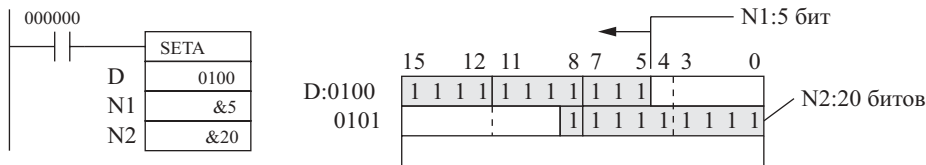
**Флаги**

Наименование	Метка	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, если N1 находится вне указанного диапазона от 0000 000F. Переводится в состояние OFF в других случаях.

**Примеры**

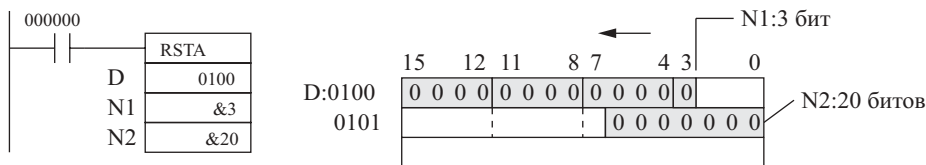
**Пример использования команды SETA(530)**

Когда в следующем ниже примере CIO 000000 переводится в состояние ON, 20 битов (0014 шестн.) начиная с бита 5 слова CIO 0100, переводятся в состояние ON.



**Пример использования команды RSTA(531)**

Когда в следующем ниже примере CIO 000000 переводится в состояние ON, 20 битов (0014 шестн.) начиная с бита 3 слова CIO 0100, переводятся в состояние OFF.



## 3-5 Команды последовательного управления

## 3-5-1 Команда завершения программы END: END(001)

**Назначение**

Обозначает конец программы

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом из циклов, когда условие выполнения находится в состоянии ON	END (001)
Указание немедленной регенерации		Не поддерживается

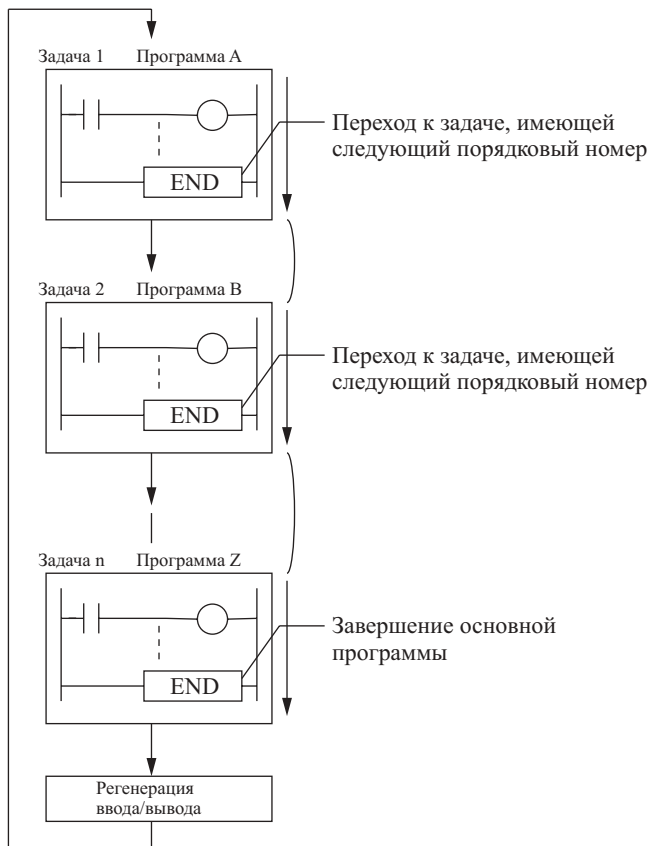
**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не используется	Не используется	Не используется	Да

**Описание**

Команда END(011) завершает выполнение программы в данном цикле. После команды END(011) ни одна из команд программы не выполняется.

Программа переходит к выполнению следующей задачи (имеющей следующий порядковый номер). Когда выполняемая задача имеет наибольший номер в данной программе, команда END(001) обозначает завершение всей основной программы.

**Меры предосторожности**

Непрерывно завершайте каждую программу командой END(001). В случае отсутствия в программе команды END(001) определяется ошибка программы.

## 3-5-2 Команда NO OPERATION: NOP(000)

**Назначение**

Данная команда не выполняет никаких функций.

**Символ релейно-контактной схемы**

Команде NOP(000) не присвоен символ релейно-контактной схемы.

**Модификации**

Модификации	Выполняется в каждом из циклов, когда условие выполнения находится в состоянии ON.	NOP(000)
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Описание**

После команды NOP(000) не выполняется никаких действий, однако эта команда может быть использована для того, чтобы оставить линии программы, куда позднее могут вводиться команды. Когда в последствии в программу вводятся команды, адреса программы не требуют введения изменений.

**Флаги**

Не существует флагов, на которые оказывают влияние команда NOP(000).

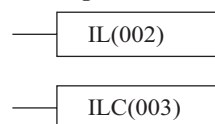
**Меры предосторожности**

Команду NOP(000) можно применять только при работе с мнемоническим дисплеем, использование ее в релейно-контактной программе не допускается.

## 3-5-3 Команды блокирования Interlock and Interlock Clear

**Назначение**

Осуществляет блокирование всех выводов, находящихся между командами IL(002) и ILC(003), когда условие выполнения команды IL(002) находится в состоянии OFF. Команды IL(002) и ILC(003) обычно используются попарно.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняет блокирование, когда условие выполнения находится в состоянии OFF/Не выполняет блокирование, когда условие выполнения находится в состоянии ON.	IL(002)
Указание немедленной регенерации		Не поддерживается
Модификации	Выполняется в каждом из циклов, когда условие выполнения находится в состоянии ON.	ILC(003)
Указание немедленной регенерации		Не поддерживается

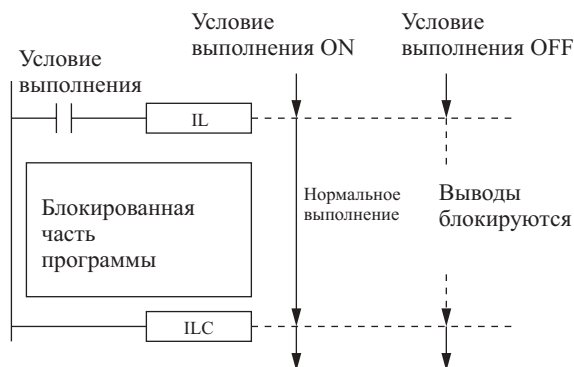
**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не используется	Не используется	Да	Да

**Описание**

Когда условие выполнения команды IL(002) находится в состоянии OFF, выходы всех команд, находящихся между командами IL(002) и ILC(003), блокируются. Когда условие выполнения команды IL(002) находится в состоянии ON, все команды, находящиеся между командами IL(002) и ILC(003), выполняются в обычном порядке.





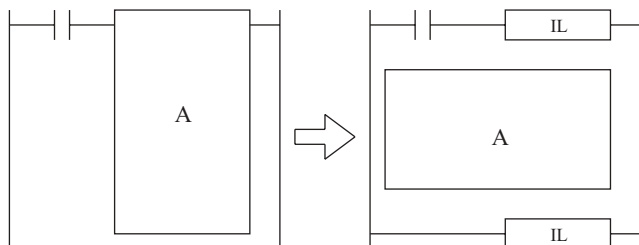
В следующей ниже таблице показано состояние различных типов выводов, находящихся в блокированной части программы, между командами IL(002) и ILC(003).

Команда	Состояние	
Биты, указанные в командах OUT и OUT NOT.	OFF	
TIM, TIMH(015), TMH(540), TIML(542)	Флаг завершения.	OFF (перезагрузка)
	Текущее значение.	Заданное значение для времени (перезагрузка)
Биты/слова, указанные во всех прочих командах.	Сохраняет предшествующее значение	

**Примечание:** Биты и слова во всех прочих командах, включая TTIM(087), MTIM(543), SET, RSET, CNT, CNTR(012), SFT, и KEEP(011) сохраняют свое состояние.

Если существуют биты, которые согласно Вашему желанию должны сохранить свое состояние ON в блокированной части программы, установите эти биты в состояние ON при помощи команды SET, вводимой непосредственно перед командой IL(002).

Переключение разделов программы с помощью команд IL(003) и ILC(003) является часто более эффективным способом. Когда несколько процессов управляются одним и тем же условием выполнения, требуется меньше шагов программы для размещения этих процессов между командами IL(002) и ILC(003).



Следующая ниже таблица показывает различие между командами IL(002) и ILC(003) и командами JMP(004) и JME(005).

Наименование	Выполнение в командах IL(002) и ILC(003)	Выполнение в командах JMP(004) и JME(005).
Выполнение команды	Выполняются команды, кроме OUT, OUT NOT и команд управления таймерами	Никакие команды не выполняются.
Состояние вывода в командах	Все выходы сохраняют свое предшествующее состояние, кроме выводов в командах OUT и OUT NOT и выводов в командах управления таймерами.	Все выходы сохраняют свое предшествующее состояние.
Биты в командах OUT и OUT NOT	OFF	Все выходы сохраняют свое предшествующее состояние.
Состояние в командах управления таймерами (кроме TTIM(087) и MTIM(543))	Перезагрузка (сброс).	Работающие таймеры (только TIM, TIMH, TMH) продолжают выполнение отсчета, т.к. текущие значения обновляются даже тогда, когда команда управления таймером не выполняется.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	OFF
Флаг отрицательного значения	N	OFF

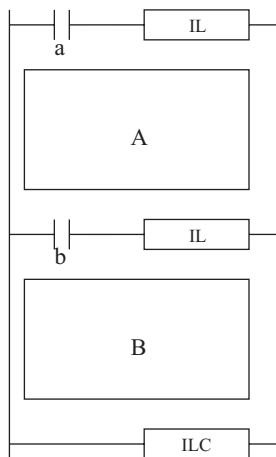
**Меры предосторожности**

Длительность цикла не сокращается, когда часть программы блокируется после выполнения внутренней команды блокирования.

Функционирование команд DIFU(013) и DIFD(014) и команд с дифференцированием зависит не только от состояния условия выполнения, когда они введены между командами IL(002) и ILC(003). Изменение условия выполнения команд DIFU(013) и DIFD(014) и команд с дифференцированием не записываются, если команды DIFU(013) и DIFD(014) находятся в заблокированной части программы и условием выполнения для IL(002) является OFF.

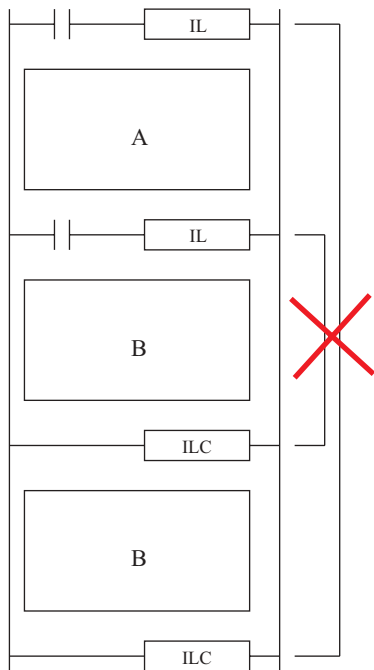
В общем случае, команды IL(002) и ILC(003) используются парами, хотя допускается использовать более одной команды IL(002) с одной командой ILC(003), как показано на следующем рисунке.

Если команды IL(002) и ILC(003) введены не попарно, при выполнении проверки программы появится сообщение об ошибке, тем не менее, программа будет выполняться должным образом.



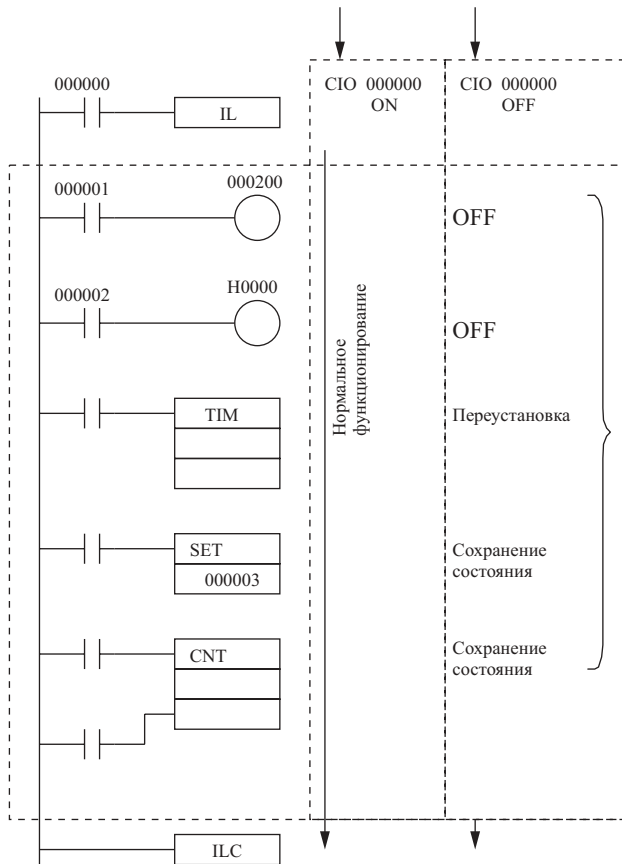
Условие выполнения		Раздел программы	
a	b	A	B
OFF	ON	Блокирован	Блокирован
OFF	OFF	Блокирован	Блокирован
ON	OFF	Не блокирован.	Блокирован
ON	ON	Не блокирован.	Не блокирован.

Команды IL(002) и ILC(003) не могут вкладываться между другими командами, как показано на следующем рисунке.



**Примеры**

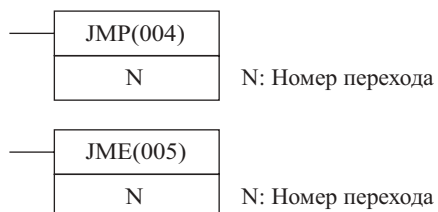
В следующем примере, когда CIO 000000 переводится в состояние OFF, все выходы, находящиеся между командами IL(002) и ILC (003), блокированы. Когда CIO 000000 переводится в состояние ON, все выходы, находящиеся между командами IL(002) и ILC(003) функционируют нормально.



## 3-5-4 Команды выполнения перехода JUMP и JUMP END: JMP(004) и JME(005)

**Назначение**

Когда условие выполнения для команды JMP(004) находится в состоянии OFF, программа выполняет переходы непосредственно к первой команде JME(005) в программе с тем же номером перехода.

**Символы релейно-контактной схемы****Модификации**

Модификации	Переход выполняется, когда условие выполнения находится в состоянии OFF/ Переход не выполняется, когда условие выполнения находится в состоянии ON.	JMP(004)
Указание немедленной регенерации		Не поддерживается
Модификации	Выполняется в каждом из циклов, когда условие выполнения находится в состоянии ON.	JME(005)
Указание немедленной регенерации		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Не используется	Да	Да

**Операнды****N: Номер перехода**

Номер перехода должен находиться в пределах от 0000 до 03FF (от 0 до 1023 в десятичном коде).

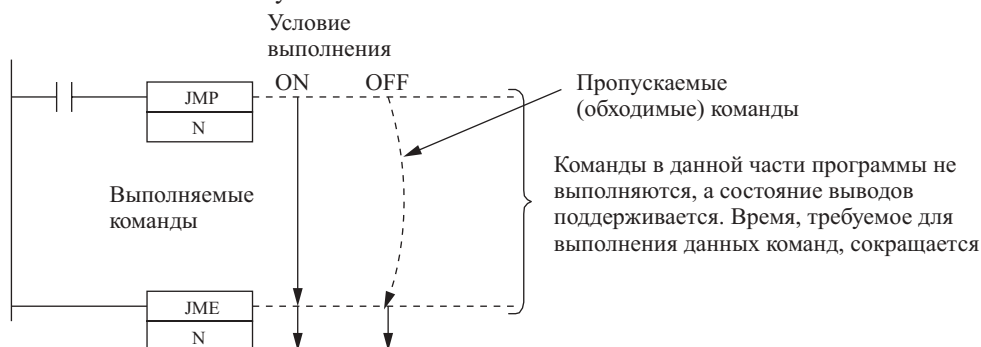
**Спецификации операндов**

Область	N	
	JMP(004)	JME(005)
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	—
Рабочая область	W000...W511	—
Область удержания бита (Holding bit area)	H000...H511	—
Область вспомогательных битов	A000...A959	—
Область таймера	T0000...T4095	—
Область счетчика	C0000...C4095	—
Область DM	D00000...D32767	—
Область EM, не содержащая банков	E00000...E32767	—
Область EM, содержащая банки	En_00000...En_32767 (n=0...C)	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)	—
Константы	#0000 - #03FF (двоичн.) или &0 &1023	#0000 - #03FF (двоичн.) или &0...&1023
Регистры данных	DR0...DR15	—
Индексные регистры	—	—
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047,IR0 -2048...+2047,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) , -(-)IR0...,(-)IR15	—

**Описание**

Когда условие выполнения команды JMP(004) находится в состоянии ON, переходы в программе не выполняются, т.е. программа выполняется последовательно, в соответствии с ее содержанием.

Когда условие выполнения для команды JMP(004) находится в состоянии OFF, программа выполняет переходы непосредственно к первой команде JME(005) с тем же номером перехода. Команды, находящиеся между JMP(004) и JME(005), не выполняются, а состояние выводов между JMP(004) и JME(005) поддерживается. В блочных программах команды, находящиеся между JMP(004) и JME(005), пропускаются независимо от состояния условия выполнения.



Вследствие того, что все команды, находящиеся между JMP(004) и JME(005) пропускаются, когда условие выполнения команды JMP(004) находится в состоянии OFF, длительность цикла уменьшается на время, требуемое для выполнения пропускаемых команд. В отличие от сказанного, при использовании команды NOP(000), длительность цикла значительно не сокращается, как при использовании команд перехода, т.к. в этом случае осуществляется выполнение команды NOP(000) для всех команд, находящихся между JMP(004) и JME(005).

В следующей ниже таблице приводится различие между командами перехода.

Наименование	JMP(004), JME(005)	CJP(510), JME(005)	CJPN(511), JME(005)	JMP0(515), JME0(516)
Условие выполнения перехода	OFF	ON	OFF	OFF
Применяемое количество	Всего 1024	Не ограничено		
Выполнение команды при переходе	Не выполняется	Выполнение команды NOP(000)		
Время выполнения команд	Нет	То же, что для команды NOP(000)		
Состояние выводов (битов и слов) при выполнении перехода	Биты и слова поддерживают свое состояние.			
Состояние работающих таймеров при выполнении перехода	Работающие таймеры продолжают отсчет.			
Действия в блочных программах	Всегда выполняется переход.	Переход выполняется при условии выполнения ON.	Переход выполняется при условии выполнения OFF.	Не разрешается

**Флаги**

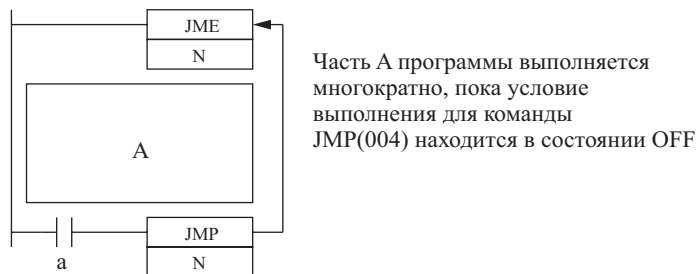
Наименование	Обозначение	Выполнение
Флаг ошибки	ER	Переводится в состояние ON, если N находится вне диапазона от 0000 до 03FF. Переводится в состояние ON, если программа содержит команду JMP(004) и не содержит команды JME(005) с тем же номером перехода. Переводится в состояние ON, если задача содержит команду JMP(004) и не содержит команды JME(005) с тем же номером перехода. Переводится в состояние OFF во всех других случаях.

**Меры предосторожности**

Все выходы (биты и слова) в пропускаемых командах сохраняют предшествующее состояние. Действующие таймеры (TIM, TIMH(015), TMNH(540)) продолжают выполнение отсчета, т.к. текущие значения таймеров обновляются даже когда команды управления таймерами не выполняются.

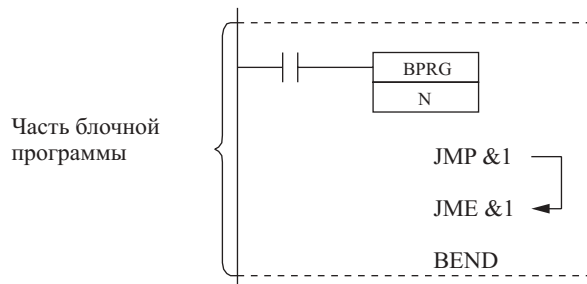
Когда программа содержит две и более команд JME(005) с одним номером перехода, действительной будет только команда с наименьшим адресом. Команда JME(005) с более высоким адресом программы игнорируется.

Когда в программе команда JME(005) предшествует команде JMP(004), команды, находящиеся между JME(005) и JMP(004), будут выполняться многократно до тех пор, пока условие выполнения для команды JMP(004) находится в состоянии OFF. Если условие выполнения не переводится в состояние ON, или команда END(001) не выполняется в течение заданной максимальной длительности цикла, определяется ошибка превышения длительности цикла.



Часть А программы выполняется многократно, пока условие выполнения для команды JMP(004) находится в состоянии OFF.

В блочных программах команды, находящиеся между JME(005) и JMP(004), обычно пропускаются, независимо от состояния условия выполнения команды JMP(004).



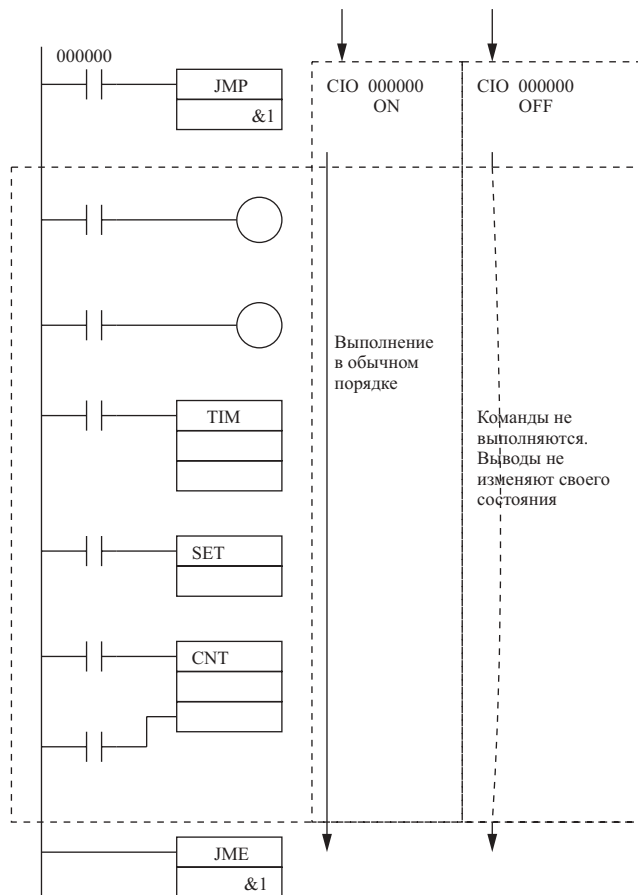
Пары, состоящие из команд JME(005) и JMP(004), должны находиться в одной задаче, так как выполнение переходов между задачами не разрешается. В случае, когда команда JME(005) вводится в другую задачу, нежели соответствующая ей команда JMP(004), определяется ошибка.

Функционирование команд DIFU(013) и DIFD(014), а также дифференцированных команд зависит не только от состояния условия выполнения, если они находятся между командами JMP(004) и JME(005). Когда команды DIFU(013) и DIFD(014), а также дифференцированные команды выполняются в обходимой части программы немедленно после того, как условие выполнения команды JMP(004) переводится в состояние ON, условие выполнения команд DIFU(013) и DIFD(014), или дифференцированных команд, сравнивается с условием выполнения, которое существовало до момента, когда переход стал действительным (т.е. до момента, когда условие выполнения команды JMP(004) перешло в состояние OFF).

### Примеры

В следующем ниже примере, когда СЮ 000000 находится в состоянии OFF, команды, находящиеся между командами JMP(004) и JME(005), не выполняются, и выходы поддерживают свое предшествующее состояние.

В следующем ниже примере, когда СЮ 000000 находится в состоянии ON, команды, находящиеся между командами JMP(004) и JME(005), выполняются в обычном порядке.



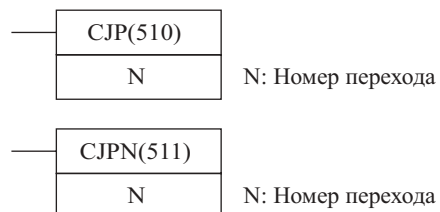
### 3-5-5 Команды условного перехода CONDITIONAL JUMP: CJP(510)/CJPN(511)

#### Назначение

Действия при выполнении команды CJP(510) противоположны действиям, выполняемым по команде JMP(004). Когда условие выполнения команды CJP(510) находится в состоянии ON, программа осуществляет переход непосредственно к первой команде JME(005) с тем же номером перехода. Команды CJP(510) и JME(005) используются парами.

Функционирование команды CJPN(511) практически аналогично команде JMP(004). Когда условие выполнения команды CJPN(511) находится в состоянии OFF, программа осуществляет переход в программе непосредственно к первой команде JME(005) с тем же номером перехода. Команды CJPN(511) и JME(005) используются парами.

#### Символы релейно-контактной схемы



#### Модификации

Модификации	Переход выполняется, когда условие выполнения находится в состоянии ON/ Переход не выполняется, когда условие выполнения находится в состоянии OFF.	CJP(510)
Указание немедленной регенерации		Не поддерживается
Модификации	Переход выполняется, когда условие выполнения находится в состоянии OFF/ Переход не выполняется, когда условие выполнения находится в состоянии ON.	CJPN(511)

Указание немедленной регенерации		Не поддерживается
Модификации	Выполняется в каждом из циклов, когда условие выполнения находится в состоянии ON.	JME(005)
Указание немедленной регенерации		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Не используется	Да	Да

**Операнды****N: Номер перехода**

Номер перехода должен находиться в пределах от 0000 до 03FF (от 0 до 1023 в десятичном коде).

**Спецификации операндов**

Область	N		
	CJP(510)	CJPN(511)	JME(005)
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		–
Рабочая область	W000...W511		–
Область удержания бита (Holding bit area)	H000...H511		–
Область вспомогательных битов	A000...A959		–
Область таймера	T0000...T4095		–
Область счетчика	C0000...C4095		–
Область DM	D00000...D32767		–
Область EM, не содержащая банков	E00000...E32767		–
Область EM, содержащая банки	En_00000...En_32767 (n=0...C)		–
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)		–
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		–
Константы	#0000...#03FF (двоичн.) или &0 &1023		#0000...#03FF (двоичн.) или &0 &1023
Регистры данных	DR0...DR15		–
Индексные регистры	–		–
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047,IR0 -2048...+2047,IR15 DR0...DR15,IR0...IR15 ,IR0(++),...,IR15(++), ,(- -)IR0...,(- -)IR15		–

**Описание**

Функционирование команд CJP(510) и CJPN(511) различается только в условиях выполнения. Команда CJP(510) осуществляет переход к первой команде JME(005) когда условие выполнения находится в состоянии ON, команда CJPN(511) выполняет аналогичную операцию, когда условие выполнения находится в состоянии OFF.

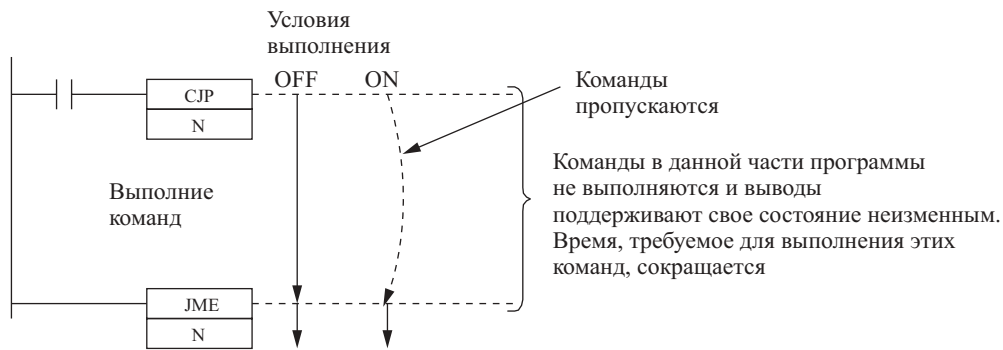
Вследствие того, что пропускаемые команды не выполняются, длительность цикла сокращается на суммарное время выполнения пропускаемых команд.

**Функционирование команды CJP(510)**

Когда условие выполнения команды CJP(510) находится в состоянии OFF, переход не осуществляется и программа выполняется последовательно, в соответствии с содержанием.

Когда условие выполнения команды CJP(510) находится в состоянии ON, программа осуществляет переход в программе к первой команде JME(005) с тем же номером перехода.





### Функционирование команды CJPN(511)

Когда условие выполнения команды CJPN(511) находится в состоянии ON, переход не осуществляется и программа выполняется последовательно, в соответствии с содержанием.

Когда условие выполнения команды CJPN(511) находится в состоянии OFF, программа осуществляет переход к первой команде JME(005) с тем же номером перехода.



### Флаги

Наименование	Обозначение	Выполнение
Флаг ошибки	ER	<p>Переводится в состояние ON, если программа содержит команду CJP(510) или CJPN(511) и не содержит команды JME(005) с тем же номером перехода.</p> <p>Переводится в состояние ON, если N находится вне указанного диапазона от 0000 до 03FF.</p> <p>Переводится в состояние ON, если задача содержит команду CJP(510) или CJPN(511) и не содержит команды JME(005) с тем же номером перехода.</p> <p>Переводится в состояние OFF во всех других случаях.</p>

### Меры предосторожности

Все выходы (биты и слова) в пропускаемых командах сохраняют предшествующее состояние. Действующие таймеры (TIM, TIMH(015), TIMN(540)) продолжают выполнение отсчета, т.к. текущие значения таймеров обновляются даже когда команды управления таймерами не выполняются.

Когда программа содержит две и более команд JME(005) с одним номером перехода, действительной будет только команда с наименьшим адресом. Команда JME(005) с более высоким адресом программы игнорируется.

Когда в программе команда JME(005) предшествует команде CJP(510) или CJPN(511), команды, находящиеся между JME(005) и CJP(510) или CJPN(511), будут выполняться многократно до тех пор, пока условие выполнения команды CJP(510) находится в состоянии OFF, или условие выполнения команды CJPN(511) находится в состоянии ON. Если переход не завершается изменением условия выполнения, или команда END(001) не выполняется в течение заданной максимальной длительности цикла, определяется ошибка превышения длительности цикла.

Выполнение команд CJP(510) и CJPN(511) в блочных программах осуществляется в обычном порядке. Когда условие выполнения команды CJP(510) находится в состоянии ON, или условие выполнения команды CJPN(511) находится в состоянии OFF, программа осуществляет переход непосредственно к команде JME(005), пропуская команды, находящиеся между CJP(510) / CJPN(511) и JME. При этом для выполнения пропускаемых команд не требуется времени, таким образом, длительность цикла сокращается.

Когда условие выполнения команды JMP0 находится в состоянии OFF, между командами JMP0 и JME0 выполняется команда NOP, требующая времени на выполнение. Следовательно, в данной ситуации длительность цикла не сокращается.

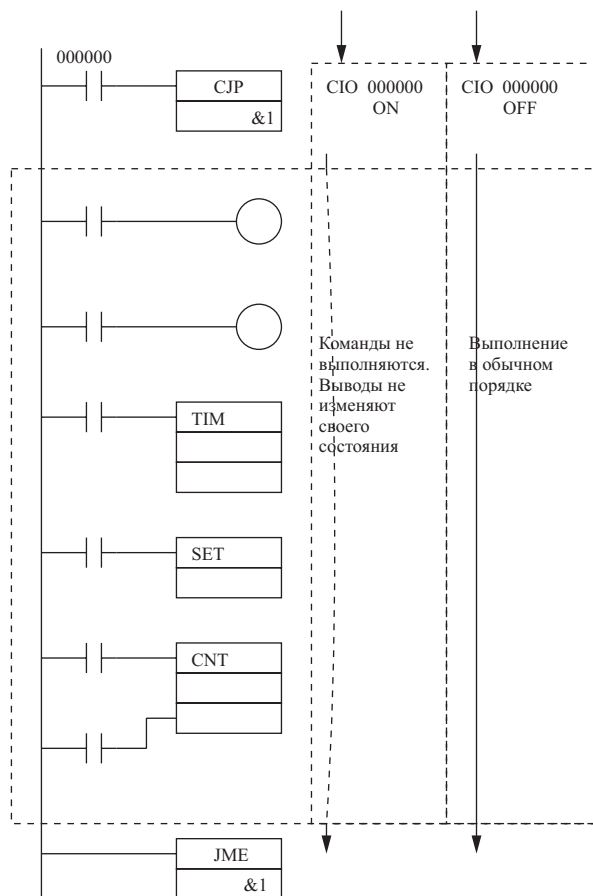
Когда в задаче присутствует команда CJP(510) или CJPN(511), в задаче должна присутствовать команда JME(005) с тем же номером перехода, так как переходы между задачами не допускаются. Если в задаче отсутствует соответствующая команда JME(005) определяется ошибка.

Функционирование команд DIFU(013) и DIFD(014), а также дифференцированных команд зависит не только от состояния условия выполнения, если они находятся в пропускаемой части программы. Когда команды DIFU(013) и DIFD(014), а также дифференцированные команды, выполняются в обходимой части программы немедленно после того, как условие выполнения команды CJP(510) переводится в состояние OFF (ON для команды CJPN(511)), условие выполнения команд DIFU(013) и DIFD(014) или дифференцированных команд, сравнивается с условием выполнения, существовавшим до момента, когда переход стал действительным.

### Примеры

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, команды, находящиеся между командами CJP(510) и JME(005), не выполняются, и выходы поддерживают свое предшествующее состояние.

Когда CIO 000000 находится в состоянии OFF, команды, находящиеся между командами CJP(510) и JME(005), выполняются в обычном порядке.

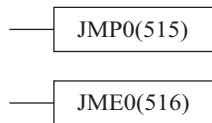


**Примечание:** Для команды CJPN(511) состояние ON/OFF CIO 000000 инвертируется.

### 3-5-6 Команды выполнения переходов MULTIPLE JUMP и JUMP END: JMP0(515) и JME(516)

#### Назначение

Когда условие выполнения команды JMP0(515) находится в состоянии ON, все команды программы, начиная от JMP0(515), и, заканчивая JME0(516) выполняются как команда NOP(000). Применяйте команды JMP0(515) и JME0(516) парами. Не существует количественных ограничений на использование в программе пар описываемых команд.

**Символы релейно-контактной схемы****Модификации**

Модификации	Переход выполняется, когда условие выполнения находится в состоянии OFF/ Переход не выполняется, когда условие выполнения находится в состоянии ON.	JMP0(515)
Указание немедленной регенерации		Не поддерживается
Модификации	Выполняется в каждом из циклов, когда условие выполнения находится в состоянии ON.	JME0(516).
Указание немедленной регенерации		Не поддерживается

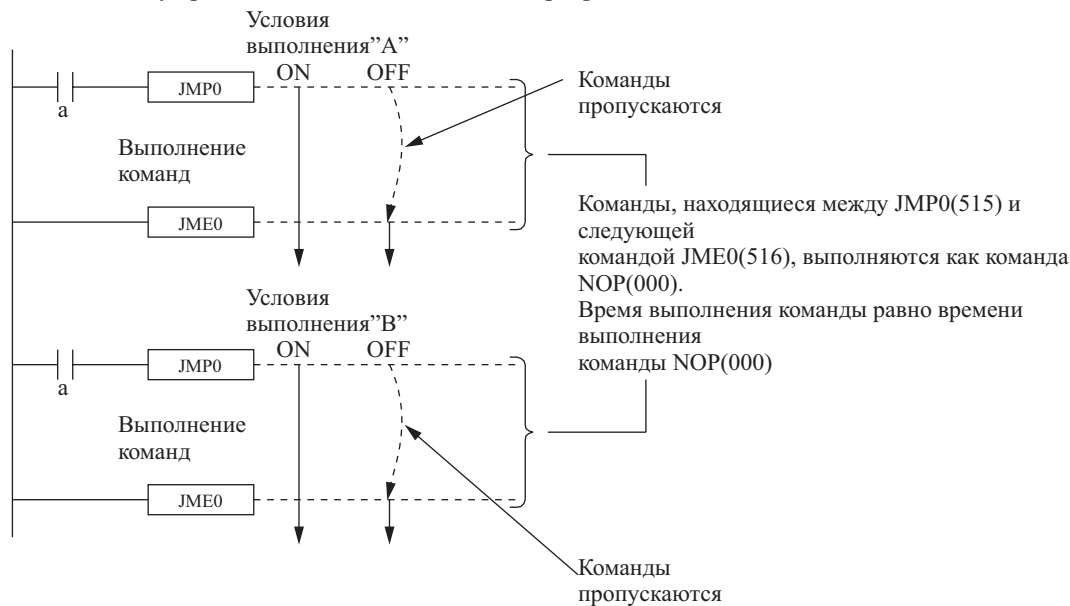
**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не используется	Не используется	Да	Да

**Описание**

Когда условие выполнения команды JMP0(515) находится в состоянии ON, переход не выполняется, и программа выполняется последовательно, согласно порядку записи команд.

Когда условие выполнения команды JMP0(515) переводится в состояние OFF, все команды, находящиеся между JMP0(515) и следующей командой JME0(516), выполняются как команда NOP(000). В отличие от команд JMP(004), CJP(510), CJPN(511), команда JMP0(515) не использует номера перехода, поэтому эти команды могут располагаться в любом месте программы.



Команды, находящиеся между JMP0(515) и следующей командой JME0(516), выполняются как команда NOP(000). Время выполнения команды равно времени выполнения команды NOP(000).

В отличие от команд JMP(004), CJP(510), CJPN(511), которые выполняют переход к первой команде JME(005) в программе, все команды, находящиеся между JMP0(515) и следующей командой JME0(516), выполняются как команда NOP(000). Время выполнения обходимых команд уменьшается, но, тем не менее, не исключается. Собственно обходимые команды не выполняются, и их выводы (слова и биты) поддерживают свое предшествующее состояние.

**Меры предосторожности**

В программе может использоваться множество парных команд JMP0(515) и JME0(516), однако не допускается образование вложений.

Когда в программе команда JME0(516) предшествует команде JMP0(515), команды, находящиеся между JME0(516) и JMP0(515), выполняются многократно, до тех пор, пока условие выполнения команды

JMP0(515) находится в состоянии OFF. Если до истечения максимального времени цикла условие выполнения не переводится в состояние ON, или не выполняется команда END(001), определяется ошибка превышения длительности цикла.

Команды JMP0(515) и JME0(516) не могут применяться в блочных программах.

Когда условие выполнения команды JMP0(515) находится в состоянии OFF, между командами JMP0 и JME0 выполняется команда NOP(000), требующая определенного времени выполнения. Учитывая сказанное, длительность цикла не сокращается.

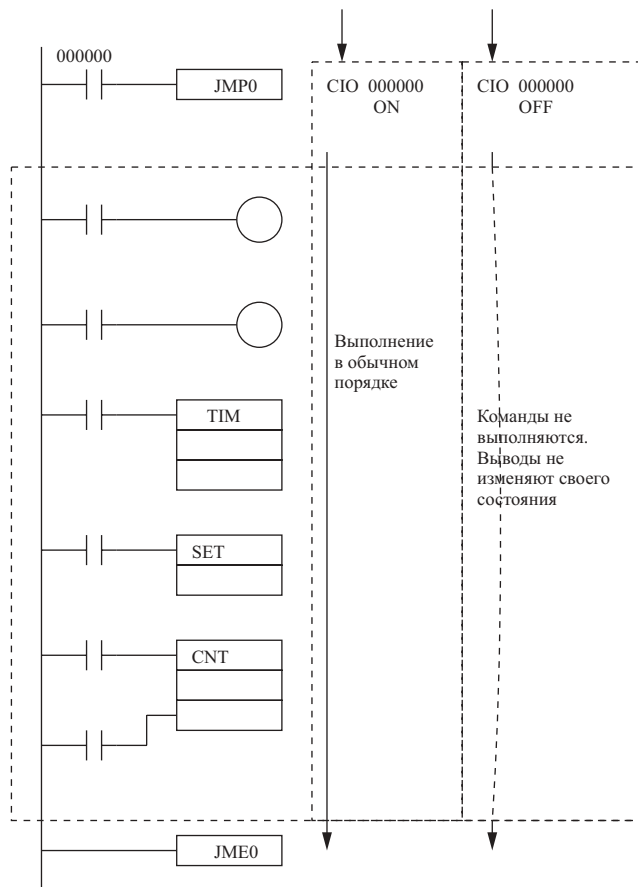
Команды JMP0(515) и JME0(516) должны находиться в одной задаче, так как переходы между задачами не допускаются. Если при использовании команды JMP0(515) в задаче отсутствует соответствующая команда JME0(516) определяется ошибка.

Функционирование команд DIFU(013) и DIFD(014), а также дифференцированных команд зависит не только от состояния условия выполнения, если они находятся в пропускаемой части программы, между командами JMP0(515) и JME0(516). Когда команды DIFU(013) и DIFD(014), а также дифференцированные команды, выполняются в обходимой части программы немедленно после того, как условие выполнения команды JMP0(515) переводится в состояние ON, условие выполнения команд DIFU(013) и DIFD(014) или дифференцированных команд, сравнивается с условием выполнения, существовавшим до момента, когда переход стал действительным (т.е. когда условие выполнения команды JMP0(515) переходит в состояние OFF).

### Примеры

В следующем ниже примере, когда CIO 000000 находится в состоянии OFF, команды, находящиеся между командами JMP0(515) и JME0(516), выполняются как команды NOP(000), а выходы поддерживают свое предшествующее состояние.

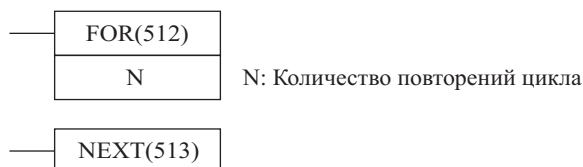
Когда CIO 000000 находится в состоянии ON, команды, находящиеся между командами JMP0(515) и JME0(516), выполняются в обычном порядке.



### 3-5-7 Циклы FOR-NEXT: FOR(512)/NEXT(513)

#### Назначение

Команды, находящиеся между FOR(512) и NEXT(513) повторяются заданное количество раз. Команды FOR(512) и NEXT(513) используются парами.

**Символы релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом из циклов, когда условие выполнения находится в состоянии ON.	FOR(512)
	Выполняется в каждом из циклов, когда условие выполнения находится в состоянии ON.	NEXT(513)
Указание немедленной регенерации		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не используется	Не используется	Да	Да

**Операнды****N: количество повторений цикла**

Количество повторений цикла должно находиться в пределах от 0000 до FFFF (от 0 до 65535 в десятичном коде).

**Спецификации операндов**

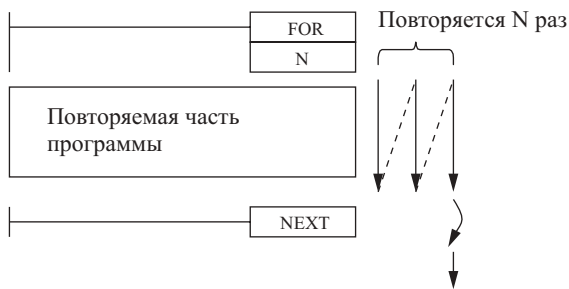
Область	N
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143
Рабочая область	W000...W511
Область удержания бита (Holding bit area)	H000...H511
Область вспомогательных битов	A000...A959
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область DM	D00000...D32767
Область EM, не содержащая банков	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)
Константы	#0000...#FFFF (двоичн.) или &0 &65535
Регистры данных	DR0...DR15
Индексные регистры	-
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047,IR0 -2048...+2047,IR15 DR0...DR15, IR0...IR15 , IR0+(++)..., IR15+(++) , -(-)IR0..., -(-)IR15

**Описание**

Команды, находящиеся между FOR(512) и NEXT(513) повторяются N раз, затем выполнение программы продолжается с команды, следующей за NEXT(513). Для отмены выполнения цикла может использоваться команда BREAK(514).

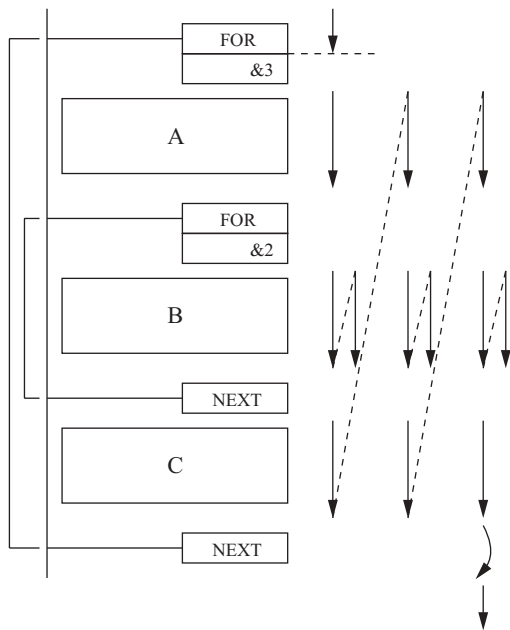
Когда N устанавливается в нулевое значение, команды, находящиеся между FOR(512) и NEXT(513) выполняются как команды NOP(000).

Циклы могут использоваться для обработки таблиц данных при минимальных затратах на программирование.



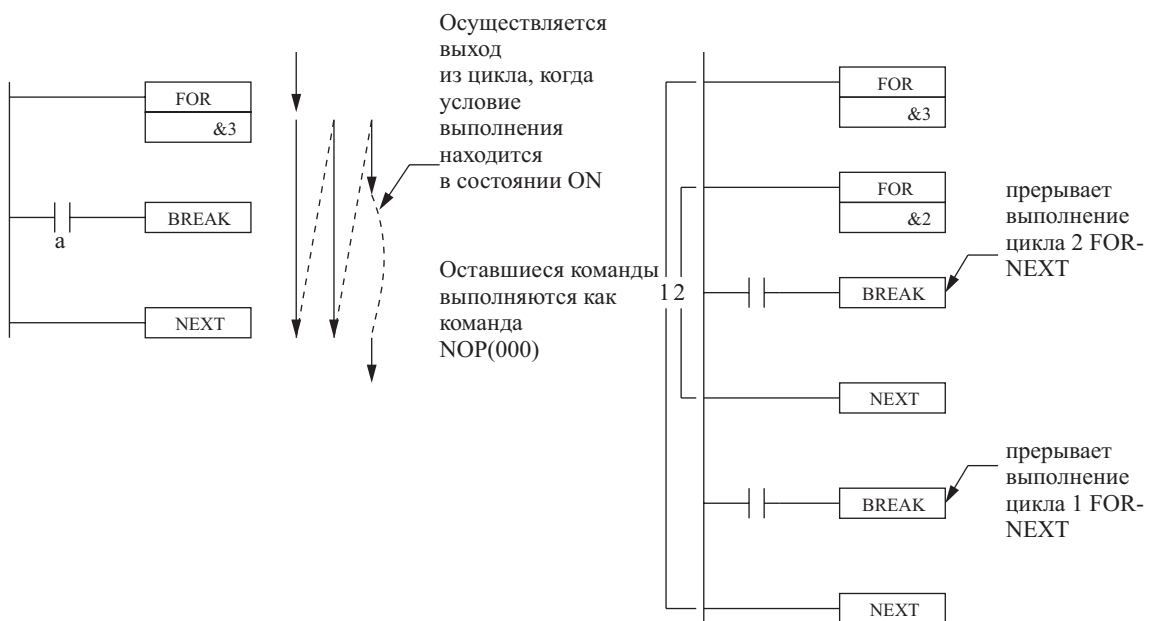
Циклы FOR-NEXT могут вкладываться один в другой, вплоть до 15 уровней. В следующем примере, части программы A, B, C выполняются следующим образом.

$A \rightarrow B \rightarrow B \rightarrow C$ ,  $A \rightarrow B \rightarrow B \rightarrow C$ , и  $A \rightarrow B \rightarrow B \rightarrow C$ .



Для выхода из цикла FOR-NEXT используйте команду BREAK(514). Для выхода из вложенных циклов требуется несколько команд BREAK(514) (соответственно количеству вложенных циклов).

Все команды цикла после команды BREAK(514) выполняются как команды NOP(000).



**Альтернативный метод создания циклов**

Существует два способа для повторения части программы до тех пор, пока на вход не поступает заданное условие выполнения.

- 1,2,3...**
1. Цикл *FOR-NEXT* с использованием команды *BREAK*.  
Запускайте цикл *FOR-NEXT* с максимальным количеством повторений *N*. Программируйте в цикле выполнение команды *BREAK(514)* при наступлении определенных условий. Выполнение цикла прерывается, не дожидаясь *N* повторений, когда на входе появляются заданные условия выполнения.
  2. Цикл с использованием команд *JME(005)* – *JMP(004)*.  
Запрограммируйте цикл, в котором команда *JME(005)*, предшествует команде *JMP(004)*. Команды, находящиеся между *JME(005)* и *JMP(004)* будут повторяться многократно, до тех пор, пока условие выполнения команды *JMP(004)* находится в состоянии *OFF*. (В случае если по истечении заданного максимального времени цикла условие выполнения не переходит в состояние *ON* или не выполняется команда *END(001)*, определяется ошибка превышения длительности цикла.)

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда произведено вложение более 15-ти циклов. Переводится в состояние OFF в других случаях.
Флаг равенства	=	OFF
Флаг отрицательного значения	N	OFF

**Меры предосторожности**

Программируйте команды *FOR(512)* и *NEXT(513)* в одной задаче. Повторение действий не осуществляется, если указанные команды находятся в разных задачах.

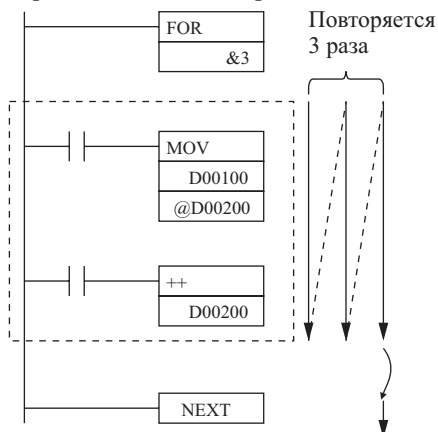
Команда выполнения перехода, например *JMP(004)*, может выполняться в цикле *FOR-NEXT*, однако переход не выполняется за пределы цикла *FOR-NEXT*.

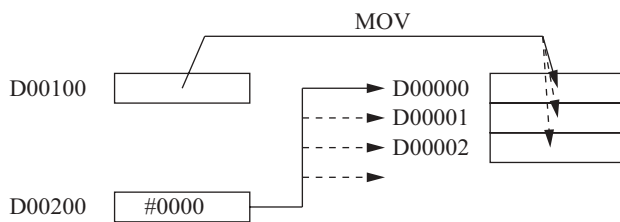
В циклах *FOR-NEXT* не допускается использование следующих ниже команд:

- Команд программирования блоков.
- Команд *MULTIPLE JUMP*, *JUMP END: JMP(515)* и *JME(516)*.
- Команд *STEP DEFINE* и *STEP START: STEP(008)* и *SNXT(009)*.

**Пример**

В следующем ниже примере часть программы, заключенная в отдельный цикл, осуществляет передачу содержания *D00100* в адрес, обозначенный в *D00200*, и затем увеличивает содержание *D00200* на единицу.





### 3-5-8 Команда отмены цикла BREAK LOOP: BREAK(514)

#### Назначение

Вводится в цикл FOR-NEXT для отмены выполнения цикла при заданном условии выполнения. Оставшиеся в цикле команды выполняются как команды NOP(000).

#### Символ релейно-контактной схемы



#### Модификации

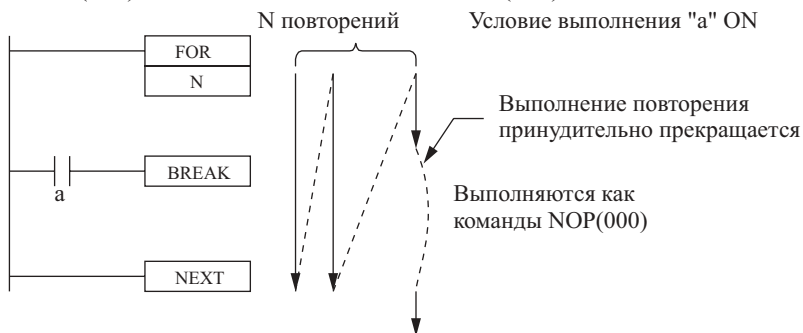
Модификации	Выполняется в каждом цикле для состояния ON.	BREAK(514).
	Выполняется один раз для дифференцирования вверх	Не поддерживается.
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается.

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не используется	Да	Да	Да

#### Описание

Для отмены выполнения цикла FOR-NEXT вводите команду BREAK(514) между командой FOR(512) и командой NEXT(513). При выполнении команды BREAK(514) прочие команды вплоть до команды NEXT(513) выполняются как команда NOP(000).



#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF.
Флаг равенства	=	OFF
Флаг отрицательного значения	N	OFF

#### Меры предосторожности

Команда BREAK(514) отменяет выполнение одного цикла, поэтому для выхода из вложенных циклов требуется несколько таких команд (соответственно количеству уровней вложения).

Команда BREAK(514) может применяться только в циклах FOR-NEXT.



## 3-6 Команды управления таймерами и счетчиками

В настоящем разделе приводится описание команд, используемых для установки таймеров и счетчиков и управления ими.

Команда	Мнемоническое обозначение	Функциональный код	Страница
TIMER	TIM	–	139
HIGH-SPEED TIMER	TIMH	015	142
ON-MS TIMER	TMNH	540	145
ACCUMULARIVE TIMER	TTIM	087	147
LONG TIMER	TIML	542	151
MULTI OUTPUT TIMER	MTIM	543	153
COUNTER	CNT	–	158
REVERSIBLE COUNTER	CNTR	012	160
RESET TIMER/COUNTER	CNR	545	164

В следующей ниже таблице приводятся основные характеристики таймеров.

Наименование	TIM	TIMH(015)	TMNH(540)	TTIM(087)	TIML(542)	MTIM(543)
Метод отсчета	Уменьшение	Уменьшение	Уменьшение	Увеличение	Уменьшение	Увеличение
Единица отсчета	0.1 сек	0.01 сек	0.001 сек	0.1 сек	0.1 сек	0.1 сек
Максимальное задаваемое значение	999.9сек	99.99 сек	9.999 сек	999.9 сек	9999999.9 сек	999.9 сек
Выводы/команда	1	1	1	1	1	8
Номера таймеров	Используется	Используется	Используется	Используется	Не используется	Не используется
Регенерация флага завершения	При выполнении	При выполнении	По прерыванию, каждую 1 мсек.	При выполнении	При выполнении	При выполнении
Регенерация текущего значения таймера	См. примечание 1	См. примечание 2	Каждую 1 мсек.	При выполнении	При выполнении	При выполнении
Значение после сброса	Флаги завершения	OFF	OFF	OFF	OFF	OFF
	Текущие значения	Заданное значение	Заданное значение	Заданное значение	0	Заданное значение

- Примечание:**
1. Текущие значения таймера (TIM PV's) подвергаются регенерации при выполнении, при завершении выполнения программы в каждом из циклов, или каждые 80 мсек. по прерыванию, если длительность цикла превышает 80 мсек.
  2. Текущие значения высокоскоростного таймера (TIMH(015)) подвергаются регенерации при выполнении, при завершении выполнения программы в каждом из циклов, или каждые 10 мсек. по прерыванию.

В следующей ниже таблице приводится влияние рабочих условий и условий программирования на функционирование таймеров.

Наименование	TIM	TIMH(015)	TMNH(540)	TTIM(087)	TIML(542)	MTIM(543)
Изменение режима работы	Текущее значение = 0 Флаг завершения = OFF			–	–	–
Прерывание подачи питания/переустановка	Текущее значение = 0 Флаг завершения = OFF			–	–	–
Выполнение CNR(545)	Текущее значение = FFFF или 9999 Флаг завершения = OFF			–	Не применяется	Не применяется
Функционирование в обходной части программы (JMP(004)-JME(005))	Действующие таймеры продолжают отсчет.			Состояние таймера поддерживается.		
Функционирование в заблокированной части программы (IL(002)-ILC(003))	Текущее значение = Заданное значение Флаг завершения = OFF			Состояние таймера поддерживается	Текущее значение = Заданное значение Флаг завершения = OFF	Состояние таймера поддерживается
Принудительная установка	Флаги завершения	ON			–	–
	Текущие значения	Установлено в значение 0			–	–

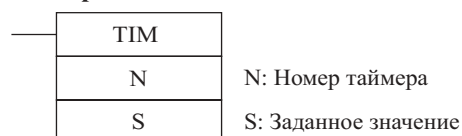
Наименование		TIM	TIMH(015)	TMNH(540)	TTIM(087)	TIML(542)	MTIM(543)
Принудительная переустановка	Флаги завершения	OFF				–	–
	Текущие значения	Переустановлено в заданное значение			Установлено в значение 0	–	–

### 3-6-1 Команда запуска таймера TIMER: TIM

#### Назначение

Команда TIM запускает таймер с отрицательным приращением текущего значения в единицах, равных 0.1 сек. Пределы установки заданного значения таймера (SV) – от 0 до 999.9 сек. Точность отсчета таймера – от 0 до 0.01 сек.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	TIM
	Выполняется один раз для дифференцирования вверх	Не поддерживается.
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не используется	Да	Да	Не используется

#### Операнды

##### N: Номер таймера

Номер таймера должен находиться в пределах от 0 до 4095.

##### S: заданное значение

Заданное значение должно находиться в пределах от 0000 до 9999.

#### Спецификации операндов

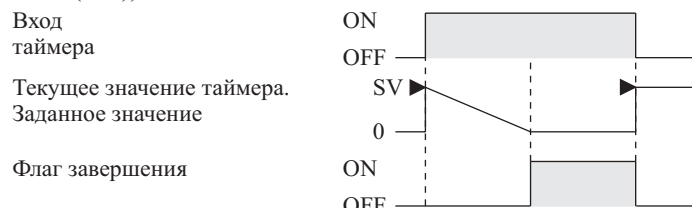
Область	N	S
Область ввода/вывода (область CIO)	–	CIO 0000...CIO 6143
Рабочая область	–	W000...W511
Область удержания бита (Holding bit area)	–	H000...H511
Область вспомогательных битов	–	A000...A959
Область таймера	T0000...T4095	T0000...T4095
Область счетчика	–	C0000...C4095
Область DM	–	D00000...D32767
Область EM, не содержащая банков	–	E00000...E32767
Область EM, содержащая банки	–	En_00000...En_32767 (n=0...C)
Область	N	S
Адреса DM/EM памяти при косвенном указании в двоичном коде	–	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)
Константы	–	0000...9999
Регистры данных	–	DR0...DR15
Индексные регистры	–	–
Косвенная адресация с использованием индексных регистров	,IR0 ...,IR15 -2048...+2047,IR0 -2048...+2047, IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(++) ,(- -)IR0...,-(- -)IR15	

**Описание**

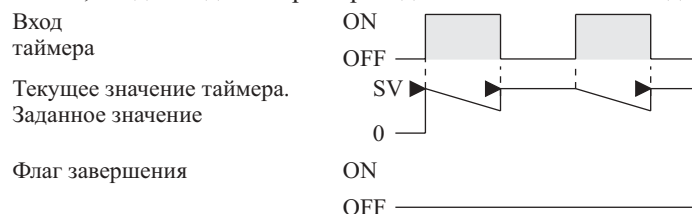
Когда на входе таймера присутствует состояние OFF, таймер, указанный числом N подвергается переустановке, т.е. текущее значение таймера сбрасывается в заданное значение, и флаг завершения переводится в состояние OFF.

Когда вход таймера переходит из состояния OFF в состояние ON, по команде TIM начинается убывающий отсчет. Текущее значение убывает по мере счета до тех пор, пока на входе таймера остается состояние ON, а флаг завершения переводится в состояние ON, когда текущее значение достигает величины, равной 0000.

Текущее значение и состояние флага завершения поддерживается после завершения отсчета. Для перезапуска таймера, его вход должен быть переведен вначале в состояние OFF, а затем в состояние ON, или текущее значение таймера должно быть переведено в значение, отличное от нуля (например, по команде MOV(021)).



На следующей ниже диаграмме показан характер изменения текущего значения и состоянии флага завершения, когда вход таймера переводится в состояние OFF до окончания отсчета.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда адрес N косвенно указывается через индексный регистр, однако адрес в индексном регистре не является адресом текущего значения таймера. Переводится в состояние ON, если S не содержит данных в двоично-десятичном коде. Переводится в состояние OFF в других случаях.
Флаг равенства	=	OFF
Флаг отрицательного значения	N	OFF

**Меры предосторожности**

Номера таймеров распределяются между командами TIM, TIMH(015), TMNH(540), TTIM(087), TIMW(813), и TMHW(815). Если один и тот же номер используется двумя таймерами, однако, таймеры не используются одновременно, при проверке программы определяется ошибка дублирования номера, тем не менее, оба таймера функционируют нормально. Таймеры, использующие один и тот же номер, не могут функционировать надлежащим образом при одновременной их работе.

Текущие значения таймеров с номерами от 0000 до 2047, обновляются даже в том случае, когда таймер находится в состоянии ожидания. Текущие значения таймеров с номерами от 2048 до 4095, удерживаются, даже когда таймер находится в состоянии ожидания.

Таймеры переустанавливаются или переводятся в состояние ожидания в следующих ниже случаях. (Когда таймер переустанавливается, его текущее значение сбрасывается в заданное значение, а флаг завершения переводится в состояние OFF.)

Условие	Текущее значение	Флаг завершения
Режим работы изменяется из рабочего режима (RUN) или режима программирования (PROGRAM) в режим мониторинга (MONITOR), или наоборот. <sup>1</sup>	0000	OFF
Прерывание подачи питания и переустановка. <sup>2</sup>	0000	OFF

Условие	Текущее значение	Флаг завершения
Выполнение команды CNR(545), RESET или TIMER/COUNTER	9999 <sup>3</sup>	OFF
Работа в заблокированной части программы (IL(002) ILC(003))	Переустанавливается в заданное значение.	OFF
Работа в обходимой части программы (JMP(004) JME(005)).	Текущее значение продолжает уменьшаться.	Сохраняет предшествующее состояние.

- Примечание:**
1. Если бит IOM Hold Bit (A50012) переведен в состояние ON, при изменении режима работы состояние флагов завершения и текущие значения таймеров сохраняются.
  2. Если бит IOM Hold Bit (A50012) переведен в состояние ON, и его состояние защищено в начальных установках Программируемого контроллера, состояние флагов завершения и текущие значения таймеров сохраняются даже при прерывании подачи питания.
  3. При выполнении команды TIM текущее значение устанавливается в заданное значение.

Когда команда TIM располагается в части программы, находящейся между командами IL(003) и ILC(003), и часть программы заблокирована, текущее значение переустанавливается в заданное значение, а флаг завершения переводится в состояние OFF.

Когда таймер TIM расположен в части программы, находящейся между командами JMP(004) и JME(005), и часть программы обходится, отсчет продолжается и текущее значение уменьшается. (Обходимая команда TIM не выполняется, однако текущее значение подвергается регенерации в каждом из циклов при завершении выполнения программы, поэтому отсчет продолжается.)

Когда таймер TIM принудительно устанавливается, соответствующий флаг завершения переводится в состояние ON, а текущее значение устанавливается в значение 0000. Когда осуществляется принудительная переустановка таймера, соответствующий флаг завершения переводится в состояние OFF, а текущее значение устанавливается в значение, равное заданному значению.

Регенерация флага завершения производится только при выполнении команды TIM, поэтому после завершения отсчета для перевода флага завершения в состояние ON может потребоваться задержка длительностью до одного цикла.

Когда для преобразования таймера в таймер другого типа с тем же номером используется оперативное редактирование (например, TIM TIMH(015) или TIM TMHH(540)), непременно осуществите сброс флага завершения. Таймер не может нормально функционировать, если флаг завершения не переустановлен. Текущее значение таймера TIM и его флаг завершения подвергаются регенерации следующими способами.

#### Таймеры с номерами 0000...2047

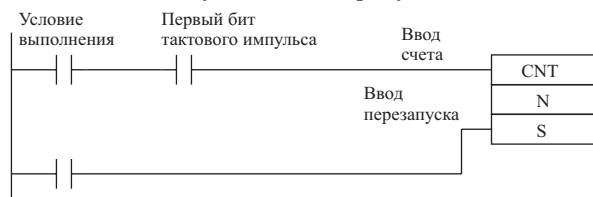
Отсчет	Регенерация
Выполнение команды TIM.	Текущее значение подвергается регенерации при каждом выполнении команды TIM. Флаг завершения переводится в состояние ON при достижении значения 0000. Флаг завершения переводится в состояние OFF, когда текущее значение не равно 0000.
Завершение выполнения программы.	Текущее значение также обновляется в каждом цикле при завершении выполнения программы.
Регенерация при прерывании.	Если длительность цикла превышает 80 мсек., текущее значение обновляется через каждые 80 мсек. по прерыванию.

#### Таймеры с номерами 2048...4096

Отсчет	Регенерация
Выполнение команды TIM.	Текущее значение подвергается регенерации при каждом выполнении команды TIM. Флаг завершения переводится в состояние ON при достижении значения 0000. Флаг завершения переводится в состояние OFF, когда текущее значение не равно 0000.

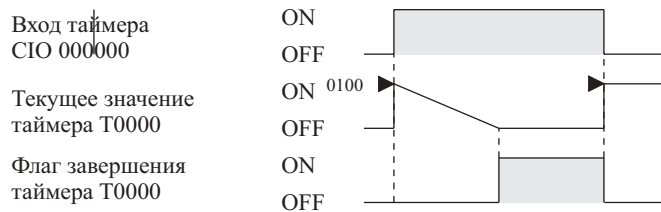
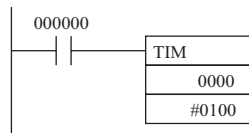
При прерывании подачи питания таймеры переустанавливаются (текущее значение изменяется на заданное значение, флаг завершения переводится в состояние OFF), если бит IOM Hold Bit (A50012) не установлен в состояние ON, и его состояние не защищено в начальных установках Программируемого

контроллера. Кроме того, существует возможность использования бита тактового импульса и команды CNT для программирования таймера, который будет сохранять свое текущее значение при прерывании подачи питания, как показано на следующем ниже рисунке.



**Пример**

В следующем ниже примере, когда ввод таймера CIO 000000 переходит из состояния OFF в состояние ON, текущее значение таймера убывает по мере отсчета, начиная с заданного значения. Флаг завершения переводится в состояние OFF, когда текущее значение достигает значения, равного 0000. Когда CIO 000000 переходит в состояние OFF, текущее значение устанавливается в заданное значение, а флаг завершения переводится в состояние OFF.

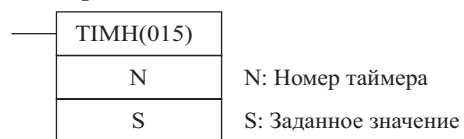


**3-6-2 Команда запуска высокоскоростного таймера HIGH-SPEED TIMER: TIMH(015)**

**Назначение**

Команда TIMH запускает таймер с отрицательным приращением текущего значения в единицах, равных 10 мсек. Пределы установки заданного значения таймера (SV) – от 0 до 99.99 сек. Точность отсчета таймера – от 0 до 0.01 сек.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	TIMH(015)
	Выполняется один раз для дифференцирования вверх.	Не поддерживается.
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не используется	Да	Да	Не используется

**Операнды**

**N: Номер таймера**

Номер таймера должен находиться в пределах от 0 до 4095.

**S: заданное значение**

Заданное значение должно находиться в пределах от 0000 до 9999 в двоично-десятичном коде.

### Спецификации операндов

Область	N	S
Область ввода/вывода (область CIO)	–	CIO 0000...CIO 6143
Рабочая область	–	W000...W511
Область удержания бита (Holding bit area)	–	H000...H511
Область вспомогательных битов	–	A000...A959
Область таймера	T0000...T4095	T0000...T4095
Область счетчика	–	C0000...C4095
Область DM	–	D00000...D32767
Область EM, не содержащая банков	–	E00000...E32767
Область EM, содержащая банки	–	En_00000...En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	–	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)
Константы	–	0000...9999
Регистры данных	–	DR0...DR15
Индексные регистры	–	...
Косвенная адресация с использованием индексных регистров	IR0 ...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15	

### Описание

Когда на входе таймера присутствует состояние OFF, таймер, указанный числом N, подвергается переустановке, т.е. текущее значение таймера сбрасывается в заданное значение, и флаг завершения переводится в состояние OFF.

Когда вход таймера переходит из состояния OFF в состояние ON, по команде TIMH(015) начинается убывающий отсчет. Текущее значение убывает по мере счета до тех пор, пока на входе таймера остается состояние ON, а флаг завершения переводится в состояние ON, когда текущее значение достигает величины, равной 0000.

Текущее значение и состояние флага завершения поддерживается после завершения отсчета. Для перезапуска таймера, его вход должен быть переведен вначале в состояние OFF, а затем в состояние ON, или текущее значение таймера должно быть переведено в значение, отличное от нуля (например, по команде MOV(021)).

На следующей ниже диаграмме показан характер изменения текущего значения и состояние флага завершения, когда вход таймера переводится в состояние OFF до окончания отсчета.

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда адрес N косвенно указывается через индексный регистр, однако адрес в индексном регистре не является адресом текущего значения таймера. Переводится в состояние ON, если S не содержит данных в двоично-десятичном коде. Переводится в состояние OFF в других случаях.
Флаг равенства	=	OFF
Флаг отрицательного значения	N	OFF

**Меры предосторожности**

Номера таймеров распределяются между командами TIM, TIMH(015), TMNH(540), TTIM(087), TIMW(813), и TMHW(815). Если один и тот же номер используется двумя таймерами, однако, таймеры не используются одновременно, при проверке программы определяется ошибка дублирования номера, тем не менее, оба таймера функционируют нормально. Таймеры, использующие один и тот же номер, не могут функционировать надлежащим образом при одновременной их работе.

Текущие значения таймеров с номерами от 0000 до 2047, обновляются даже в том случае, когда таймер находится в состоянии ожидания. Текущие значения таймеров с номерами от 2048 до 4095, удерживаются, когда таймер находится в состоянии ожидания.

Таймеры переустанавливаются или их работа прерывается в следующих ниже случаях. (Когда таймер переустанавливается, его текущее значение сбрасывается в заданное значение, а флаг завершения переводится в состояние OFF.)

Условие	Текущее значение	Флаг завершения
Режим работы изменяется из рабочего режима (RUN) или режима программирования (PROGRAM) в режим мониторинга (MONITOR), или наоборот. <sup>1</sup>	0000	OFF
Прерывание подачи питания и переустановка. <sup>2</sup>	0000	OFF
Выполнение команды CNR(545), RESET или TIMER/COUNTER	9999 <sup>3</sup>	OFF
Работа в заблокированной части программы (IL(002) ILC(003))	Переустанавливается в заданное значение.	OFF
Работа в обходимой части программы (JMP(004) JME(005)).	Текущее значение продолжает уменьшаться.	Сохраняет предшествующее состояние.

- Примечание:**
1. Если бит IOM Hold Bit (A50012) переведен в состояние ON, при изменении режима работы состояние флагов завершения и текущие значения таймеров сохраняются.
  2. Если бит IOM Hold Bit (A50012) переведен в состояние ON, и его состояние защищается в начальных установках Программируемого контроллера, состояние флагов завершения и текущие значения таймеров сохраняются даже при прерывании подачи питания
  3. При выполнении команды TIMH(015) текущее значение устанавливается в заданное значение.

Когда работающий таймер TIMH(015) расположен в части программы, находящейся между командами JMP(004) и JME(005), и часть программы обходится, отсчет продолжается и текущее значение уменьшается. (Обходимая команда TIMH(015) не выполняется, однако текущее значение подвергается регенерации в каждом из циклов при завершении выполнения программы, поэтому отсчет продолжается.)

Когда команда TIMH(015) располагается в части программы, находящейся между командами IL(003) и ILC(003), и часть программы заблокирована, текущее значение переустанавливается в заданное значение, а флаг завершения переводится в состояние OFF.

Когда таймер TIMH(015) принудительно устанавливается, соответствующий флаг завершения переводится в состояние ON, а текущее значение устанавливается в значение 0000. Когда осуществляется принудительная переустановка таймера, соответствующий флаг завершения переводится в состояние OFF, а текущее значение переустанавливается в значение, равное заданному значению.

Регенерация флага завершения осуществляется только при выполнении команды TIMH(015), поэтому после завершения отсчета для перевода флага завершения в состояние ON может потребоваться задержка длительность до одного цикла.

Когда для преобразования таймера в таймер другого типа с тем же номером используется оперативное редактирование (например, TIMH(015) TIM или TIMH(015) TMNH(540)), непременно осуществите сброс флага завершения. Таймер не может нормально функционировать, если флаг завершения не переустановлен.

Текущее значение таймера TIMH(015) и его флаг завершения подвергаются регенерации следующими способами.

**Таймера с номерами 0000...0255**

Отсчет	Регенерация
Выполнение команды TIMH(015).	Флаг завершения переводится в состояние ON при достижении значения 0000. Флаг завершения переводится в состояние OFF, когда текущее значение не равно 0000.
Регенерация с интервалом 10 мс	Текущее значение обновляется через каждые 10 мс.



**Таймера с номерами 0256...2047**

Отсчет	Регенерация
Выполнение команды ТИМН(015).	Текущее значение подвергается регенерации при каждом выполнении команды ТИМН(015). Флаг завершения переводится в состояние ON при достижении значения 0000. Флаг завершения переводится в состояние OFF, когда текущее значение не равно 0000.
Завершение выполнения программы.	Текущее значение также обновляется в каждом цикле при завершении выполнения программы.
Регенерация с интервалом 80 мс	Если время цикла превышает 80 мс, то текущее значение обновляется через каждые 80 мсек

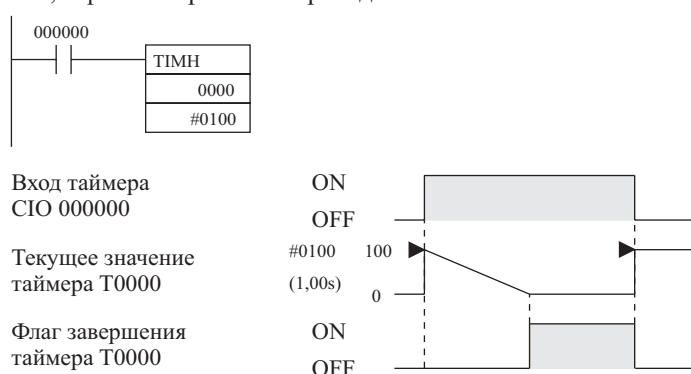
**Таймера с номерами 2048...4096**

Отсчет	Регенерация
Выполнение команды ТИМН(015).	Текущее значение подвергается регенерации при каждом выполнении команды ТИМН(015). Флаг завершения переводится в состояние ON при достижении значения 0000. Флаг завершения переводится в состояние OFF, когда текущее значение не равно 0000.

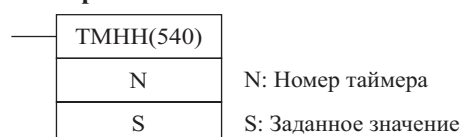
**Пример**

В следующем ниже примере, когда ввод таймера CIO 000000 переходит из состояния OFF в состояние ON, текущее значение таймера начинает убывать по мере отсчета, начиная с заданного значения (#0064 = 100 = 1.00 сек). Флаг завершения T0000 переводится в состояние ON, когда текущее значение достигает значения, равного 0000.

Когда CIO 000000 переходит в состояние OFF, текущее значение переустанавливается в заданное значение, а флаг завершения переводится в состояние OFF.

**3-6-3 Команда запуска миллисекундного таймера ONE-MS TIMER: ТМНН****Назначение**

Команда ТМНН(540) запускает таймер с отрицательным приращением текущего значения в единицах, равных 1мсек. Пределы установки заданного значения таймера (SV) – от 0 до 9.999 сек. Точность отсчета таймера – от 0 до 0.001 сек.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	ТМНН(540)
	Выполняется один раз для дифференцирования вверх.	Не поддерживается.
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.



Указание немедленной регенерации.	Не поддерживается
-----------------------------------	-------------------

**Применяемые области программы**

Области блоков программы	Области шагов программы	Подпрограммы	Задачи прерывания
Не используется	Да	Да	Не используется

**Операнды****N: Номер таймера**

Номер таймера должен находиться в пределах от 0 до 4095.

**S: заданное значение**

Заданное значение должно находиться в пределах от 0000 до 9999.

**Спецификации операндов**

Область	N	S
Область ввода/вывода (область CIO)	–	CIO 0000...CIO 6143
Рабочая область	–	W000...W511
Область удержания бита (Holding bit area)	–	H000...H511
Область вспомогательных битов	–	A000...A959
Область таймера	T0000...T0015	T0000...T4095
Область счетчика	–	C0000...C4095
Область DM	–	D00000...D32767
Область EM, не содержащая банков	–	E00000...E32767
Область EM, содержащая банки	–	En_00000...En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	–	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)
Константы	–	0000...9999
Регистры данных	–	DR0...DR15
Индексные регистры	–	–
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15	

**Описание**

Когда на входе таймера присутствует состояние OFF, таймер, указанный числом N, подвергается переустановке, т.е. текущее значение таймера сбрасывается в заданное значение, и флаг завершения переводится в состояние OFF.

Когда вход таймера переходит из состояния OFF в состояние ON, по команде ТМНН(540) начинается убывающий отсчет. Текущее значение убывает по мере счета до тех пор, пока на входе таймера остается состояние ON, а флаг завершения переводится в состояние ON, когда текущее значение достигает величины, равной 0000.

Текущее значение и состояние флага завершения поддерживается после завершения отсчета. Для перезапуска таймера, его вход должен быть переведен вначале в состояние OFF, а затем в состояние ON, или текущее значение таймера должно быть переведено в значение, отличное от нуля (например, по команде MOV(021)).

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда адрес N косвенно указывается через индексный регистр, однако адрес в индексном регистре не является адресом текущего значения таймера. Переводится в состояние ON, если S не содержит данных в двоично-десятичном коде. Переводится в состояние OFF в других случаях.
Флаг равенства	=	OFF
Флаг отрицательного значения	N	OFF

**Меры предосторожности**

Номера таймеров распределяются между командами TIM, TIMH(015), TMNH(540), TTIM(087), TIMW(813), и TMHW(815). Если один и тот же номер используется двумя таймерами, однако, таймеры не используются одновременно, при проверке программы определяется ошибка дублирования номера, тем не менее, оба таймера функционируют нормально. Таймеры, использующие один и тот же номер, не могут функционировать надлежащим образом при одновременной их работе.

Текущие значения таймеров с номерами от 0000 до 2047, обновляются даже в том случае, когда таймер находится в состоянии ожидания. Текущие значения таймеров с номерами от 2048 до 4095, удерживаются, когда таймер находится в состоянии ожидания.

Таймеры переустанавливаются или переводятся в состояние ожидания в следующих ниже случаях. (Когда таймер переустанавливается, его текущее значение сбрасывается в заданное значение, а флаг завершения переводится в состояние OFF.)

Условие	Текущее значение	Флаг завершения
Режим работы изменяется из рабочего режима (RUN) или режима программирования (PROGRAM) в режим мониторинга (MONITOR), или наоборот. <sup>1</sup>	0000	OFF
Прерывание подачи питания и переустановка <sup>2</sup>	0000	OFF
Выполнение команды CNR(545), RESET или TIMER/COUNTER	9999 <sup>3</sup>	OFF
Работа в заблокированной части программы (IL(002) ILC(003))	Переустанавливается в заданное значение.	OFF
Работа в обходимой части программы (JMP(004) JME(005)).	Текущее значение продолжает уменьшаться.	Сохраняет предшествующее состояние.

- Примечание:**
1. Если бит IOM Hold Bit (A50012) переведен в состояние ON, при изменении режима работы состояние флагов завершения и текущие значения таймеров сохраняются.
  2. Если бит IOM Hold Bit (A50012) переведен в состояние ON, и его состояние защищено в начальных установках Программируемого контроллера, состояние флагов завершения и текущие значения таймеров сохраняются даже при прерывании подачи питания.
  3. При выполнении команды TMNH(540) текущее значение устанавливается в заданное значение.

Когда таймер TMNH(540) расположен в части программы, находящейся между командами JMP(004) и JME(005), и часть программы обходится, отсчет продолжается и текущее значение уменьшается. Обходимая команда TMNH(015) не выполняется, и данные не могут обновляться при выполнении команды. Вместо этого значение подвергается регенерации в каждом из циклов при завершении программы.

Когда команда TMNH(540) располагается в части программы, находящейся между командами IL(003) и ILC(003), и программа заблокирована, текущее значение переустанавливается в заданное значение, а флаг завершения переводится в состояние OFF.

В случае, когда таймер TMNH(540) принудительно устанавливается, соответствующий флаг завершения переводится в состояние ON, а текущее значение устанавливается в значение 0000. Когда осуществляется принудительная переустановка таймера, соответствующий флаг завершения переводится в состояние OFF, а текущее значение переустанавливается в значение, равное заданному значению.

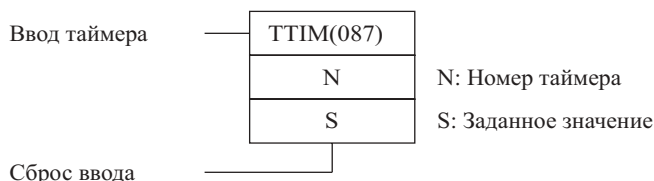
Когда для преобразования таймера в таймер другого типа с тем же номером используется оперативное редактирование (например, TMNH(540) TIM или TMNH(540) TIMH(015)), непременно осуществите сброс флага завершения. Таймер не может нормально функционировать, если флаг завершения не переустановлен.

Текущее значение таймера TMNH(540) и его флаг завершения подвергаются регенерации периодически, через интервал, равный 1 мсек. по прерыванию.

Отсчет	Регенерация
Регенерация при прерывании.	Текущее значение обновляется периодически через 1 мсек. по прерыванию.

**3-6-4 Команда запуска аккумулятивного таймера ACCUMULATIVE TIMER: TTIM(087)****Назначение**

Команда TTIM(087) запускает таймер с приращением текущего значения в единицах, равных 0.1 сек. Пределы установки заданного значения таймера (SV) – от 0 до 999.9 сек. Точность отсчета таймера – от 0 до 0.01 сек.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	TTIM(087)
	Выполняется один раз для дифференцирования вверх	Не поддерживается.
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не используется	Да	Да	Не используется

**Операнды****N: Номер таймера**

Номер таймера должен находиться в пределах от &0 до &4095 в десятичном коде.

**S: заданное значение**

Заданное значение должно находиться в пределах от 0000 до 9999.

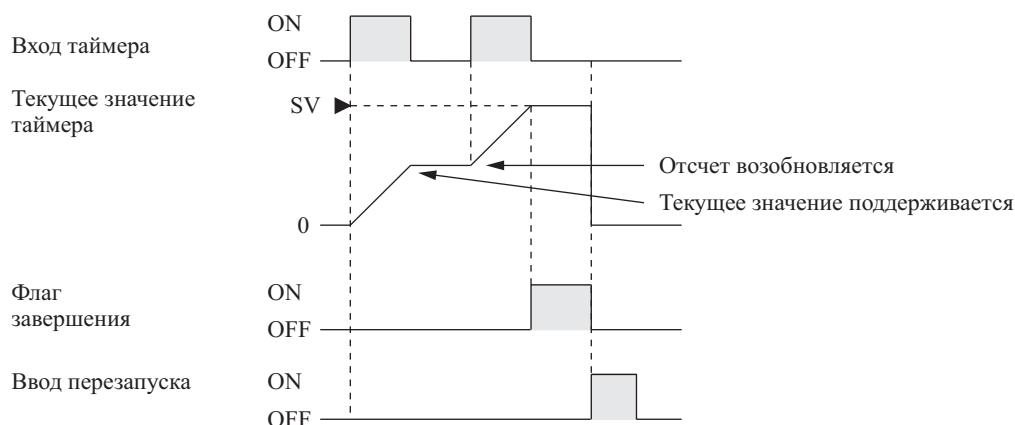
**Спецификации операндов**

Область	N	S
Область ввода/вывода (область CIO)	—	CIO 0000 CIO 6143
Рабочая область	—	W000 W511
Область удержания бита (Holding bit area)	—	H000 H511
Область вспомогательных битов	—	A000 A959
Область таймера	T0000...T4095	T0000 T4095
Область счетчика	—	C0000 C4095
Область DM	—	D00000 D32767
Область EM, не содержащая банков	—	E00000 E32767
Область EM, содержащая банки	—	En_00000 En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	—	@D00000 @D32767 @E00000 @E32767 @En_00000 @En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—	*D00000 *D32767 *E00000 *E32767 *En_00000 *En_32767 (n=0...C)
Константы	—	0000 9999
Регистры данных	—	DR0 DR15
Индексные регистры	—	---
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15	

**Описание**

Когда вход таймера переводится в состояние ON, таймер TTIM(087) осуществляет отсчет с приращением текущего значения. Когда ввод переводится в состояние OFF, таймер прекращает отсчет, однако текущее значение сохраняется. Отсчет и увеличение текущего значения возобновляется с точки прерывания, когда ввод вновь переводится в состояние ON. Когда текущее значение достигает заданной величины, флаг завершения переводится в состояние OFF.

Текущее значение и состояние флага завершения поддерживается после завершения отсчета. Для перезапуска таймера существует три способа: текущее значение таймера должно быть переведено в значение, отличное от нуля (например, по команде MOV(021)), ввод перезапуска должен быть переведен в состояние ON, или должна быть выполнена команда CNR(545).



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда адрес N косвенно указывается через индексный регистр, однако адрес в индексном регистре не является адресом текущего значения таймера. Переводится в состояние ON, если S не содержит данных в двоично-десятичном коде. Переводится в состояние OFF в других случаях.

### Меры предосторожности

Номера таймеров распределяются между командами TIM, TIMH(015), TMNH(540), TTIM(087), TIMW(813), и TMHW(815). Если один и тот же номер используется двумя таймерами, однако, таймеры не используются одновременно, при проверке программы определяется ошибка дублирования номера, тем не менее, оба таймера функционируют нормально. Таймеры, использующие один и тот же номер, не могут функционировать надлежащим образом при одновременной их работе.

Таймеры переустанавливаются или переводятся в состояние ожидания в следующих ниже случаях. (Когда таймер TTIM(087) переустанавливается, его текущее значение сбрасывается в значение, равное 0000, а флаг завершения переводится в состояние OFF.)

Условие	Текущее значение	Флаг завершения
Режим работы изменяется из рабочего режима (RUN) или режима программирования (PROGRAM) в режим мониторинга (MONITOR), или наоборот. <sup>1</sup>	0000	OFF
Прерывание подачи питания и переустановка. <sup>2</sup>	0000	OFF
Выполнение команды CNR(545), RESET или TIMER/COUNTER	9999 <sup>3</sup>	OFF
Работа в заблокированной части программы (IL(002) ILC(003))	Переустанавливается в заданное значение.	OFF
Работа в обходимой части программы (JMP(004) JME(005)).	Текущее значение продолжает уменьшаться.	Сохраняет предшествующее состояние.

- Примечание:**
1. Если бит IOM Hold Bit (A50012) переведен в состояние ON, при изменении режима работы состояние флагов завершения и текущие значения таймеров сохраняются.
  2. Если бит IOM Hold Bit (A50012) переведен в состояние ON, и его состояние защищено в начальных установках Программируемого контроллера, состояние флагов завершения и текущие значения таймеров сохраняются даже при прерывании подачи питания.
  3. При выполнении команды TTIM(087) текущее значение устанавливается в заданное значение.

Когда команда TTIM(087) располагается в части программы, находящейся между командами IL(003) и ILC(003), и часть программы заблокирована, текущее значение сохраняет свое предшествующее значение (таймер не переустанавливается). При расположении таймера TTIM(087) между командами IL(003) и ILC(003), принимайте этот факт во внимание.

Когда таймер TTIM(087) расположен в части программы, находящейся между командами JMP(004) и JME(005), и часть программы обходится, текущее значение удерживается. При расположении таймера TTIM(087) между командами JMP(004) и JME(005), принимайте этот факт во внимание.

Когда таймер ТИМН(015) принудительно устанавливается, соответствующий флаг завершения переводится в состояние ON, а текущее значение сбрасывается в значение 0000. Когда осуществляется принудительная переустановка таймера, соответствующий флаг завершения переводится в состояние OFF, а текущее значение устанавливается в значение, равное 0000. Операции принудительной установки и принудительной переустановки обладают приоритетом над состоянием входа таймера и состоянием ввода переустановки.

Регенерация текущего значения осуществляется только при выполнении команды ТТИМ(087), поэтому, когда длительность цикла превышает 100 мсек, таймер не может нормально функционировать т.к. приращения таймера выполняются в единицах, равных 100 мсек.

Регенерация флага завершения осуществляется только при выполнении команды ТТИМ(087), поэтому после завершения отсчета для перевода флага завершения в состояние ON может потребоваться задержка длительность до одного цикла.

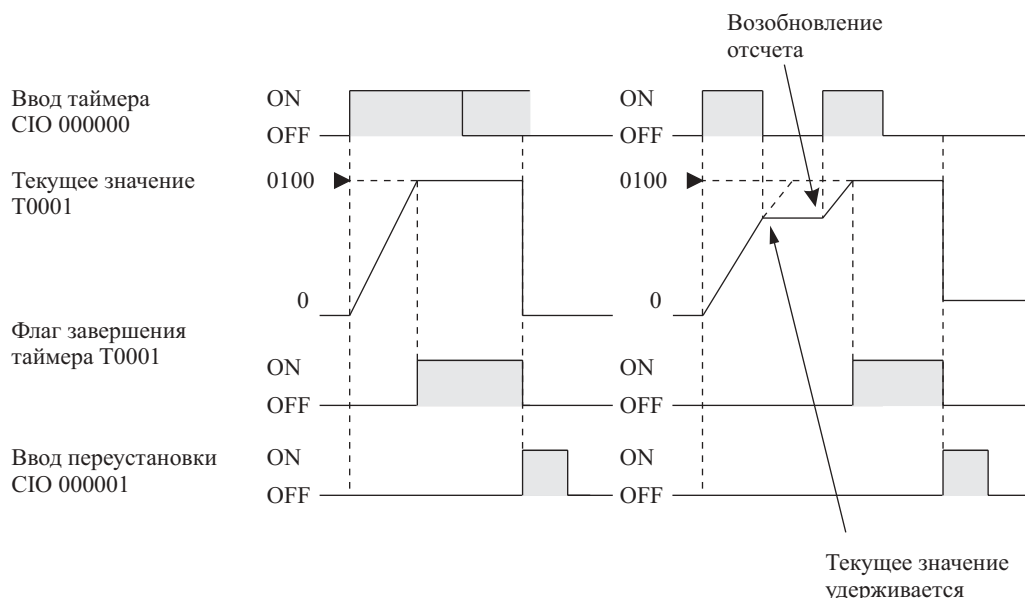
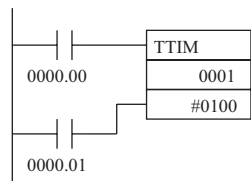
Обычные таймеры, каким является таймер ТИМ, являются таймерами с отрицательным приращением, поэтому текущее значение указывает время, оставшееся до завершения отсчета. Таймер ТТИМ(087) показывает время, прошедшее с начала отсчета, поэтому его текущее значение может использоваться при множестве вычислений и в дисплеях.

### Пример

В следующем ниже примере, когда ввод таймера СЮ 000000 переходит в состояние ON, текущее значение таймера увеличивается, начиная с нулевого значения. Флаг завершения T0001 переводится в состояние ON, когда текущее значение достигает заданного значения.

Когда ввод перезапуска переводится в состояние ON, текущее значение таймера устанавливается в значение, равное 0000, а флаг завершения (T0001) переводится в состояние OFF. (В обычном случае ввод перезапуска переводится в состояние ON для переустановки таймера, и затем ввод таймера переводится в состояние ON для его запуска.)

Если ввод таймера переводится в состояние OFF до достижения заданного значения, отсчет прерывается, однако текущее значение удерживается. Когда ввод таймера вновь переводится в состояние ON, таймер возобновляет отсчет, начиная с удерживаемого значения.



## 3-6-5 Команда запуска долговременного таймера LONG TIMER: TIML(542)

**Назначение**

Команда TIML(542) запускает таймер с отрицательным приращением текущего значения в единицах, равных 1 сек., который способен вести отсчет до 115 дней (от 0 до 9999999.9 сек). Точность отсчета таймера – от 0 до 0.01 сек.

**Символ релейно-контактной схемы**

TIML(542)	
D1	D1: Флаг завершения
D2	D2: Слово, содержащее текущее значение
S	S: Слово содержащее заданное значение

**Модификации**

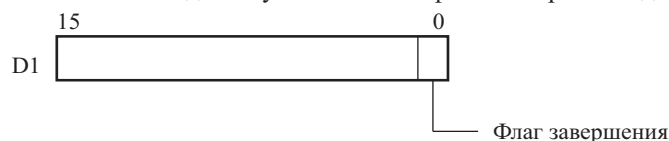
Модификации	Выполняется в каждом цикле для состояния ON.	TIML(542),
	Выполняется один раз для дифференцирования вверх	Не поддерживается.
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не используется	Да	Да	Не используется

**Операнды****D1: Флаг завершения**

Бит 0 слова D1 действует в качестве флага завершения для таймера TIML(542).

**D2: Слово, содержащее текущее значение**

D2+1 и D2 содержат восьмизначное число в двоичном или двоично-десятичном коде, равное текущему значению. (D2 и D2+1 должны находиться в одной области данных.) Диапазон изменения текущего значения от 00000000 до 99999999.

**S: слово, содержащее заданное значение**

S+1 и S содержат восьмизначное число в двоичном или двоично-десятичном коде, равное заданному значению. (S+1 и S должны находиться в одной области данных.) Диапазон установки заданного значения от 00000000 до 99999999 в двоично-десятичном коде.

**Спецификации операндов**

Область	D1	D2	S
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6142	
Рабочая область	W000...W511	W000...W510	
Область удержания бита (Holding bit area)	H000...H511	H000...H510	
Область вспомогательных битов	A000...A959	A000...A958	A000...A958
Область таймера	T0000...T4095	–	T0000...T4094
Область счетчика	C0000...C4095	–	C0000...C4094
Область DM	D00000...D32767	D00000...D32766	
Область EM, не содержащая банков	E00000...E32767	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32766 (n = 0...C)	



Область	D1	D2	S
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	–	0000...9999	
Регистры данных	–		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0...IR15		

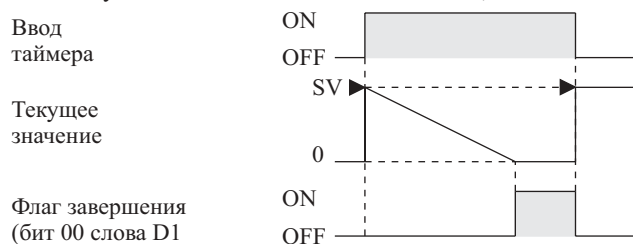
### Описание

Таймер TIML(543) представляет собой\*\*\* таймер с отрицательным приращением с единицами отсчета, равными 0.1 сек. Так как таймер использует восьмизначные заданные значения и восьмизначные текущие значения, максимальное значение отсчета достигает 999999,9 сек. (примерно 115 дней) в двоично-десятичном коде.

Когда на входе таймера присутствует состояние OFF, осуществляется переустановка таймера, т.е. текущее значение таймера устанавливается в заданное значение, а флаг завершения устанавливается в состояние OFF.

При переводе ввода из состояния OFF в состояние ON, таймер TIML(542) начинает убывающий отсчет текущего значения, содержащийся в словах D2+1 и D2. Текущее значение уменьшается, пока на входе существует состояние ON, и флаг завершения переводится в состояние ON, при достижении таймером значения 00000000.

Текущее значение и состояние флага завершения поддерживается после завершения отсчета. Для перезапуска таймера необходимо перевести вход вначале в состояние OFF, а затем в состояние ON, или изменить текущее значение на любое значение, отличное от нуля (например, с помощью команды MOV(021)).



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда текущее значение в словах D2+1 и D2 выражено не в двоично-десятичном коде. Переводится в состояние ON, если текущее значение в словах S+1 и S выражено не в двоично-десятичном коде. Переводится в состояние OFF в других случаях.

### Меры предосторожности

В отличие от других таймеров, таймер TIML(542) не использует номера. (Для таймера TIML(542) операция регенерации текущего значения области таймера не выполняется.)

Так как флаг завершения таймера TIML(542) находится в области данных, его состояние может принудительно изменяться, подобно другим битам, однако текущее значение не изменяется.

Регенерация текущего значения таймера осуществляется только при выполнении команды TIML(542), поэтому таймер не будет работать надлежащим образом, если длительность цикла превышает 100 мсек., т.к. единица приращения таймера равна 100 мсек.

Регенерация состояния флага завершения осуществляется только при выполнении команды TIML(542), поэтому для перевода флага в состояние ON после завершения отсчета может потребоваться задержка, длительностью до одного цикла.

Когда команда TIML(542) располагается в части программы, находящейся между командами IL(003) и LC(003), и программа заблокирована, текущее значение переустанавливается в заданное значение, а флаг завершения переводится в состояние OFF.

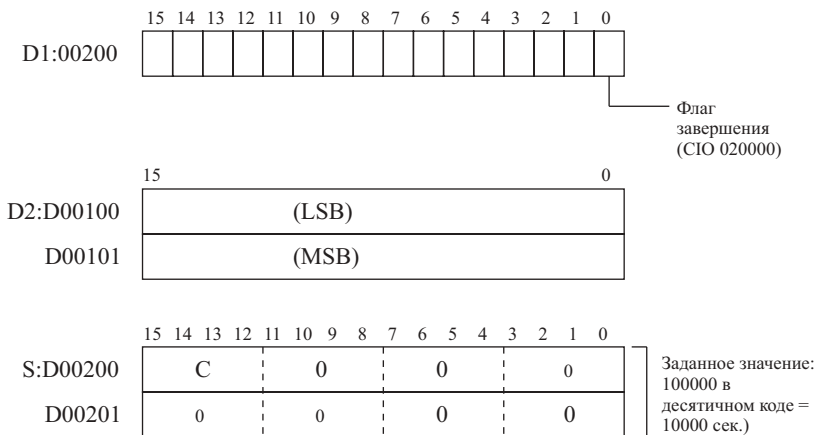
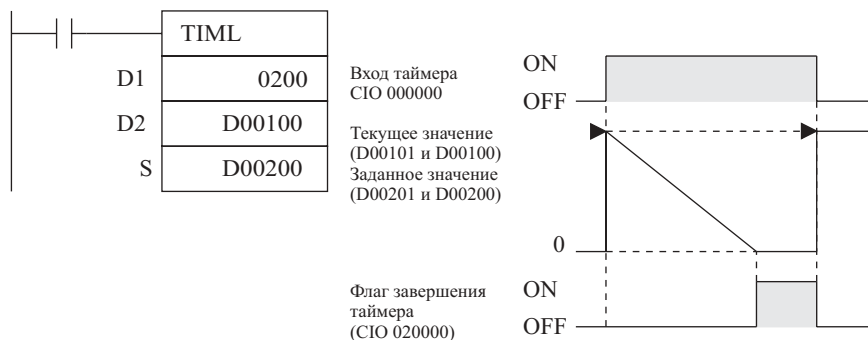
Когда таймер TIML(542) расположен в части программы, находящейся между командами JMP(004) и JME(005), и часть программы обходится, текущее значение сохраняет свое предшествующее значение. При расположении таймера TTIM(087) между командами JMP(004) и JME(005), принимайте этот факт во внимание.

Убедитесь в том, что слова, указанные в качестве флага завершения и текущего значения (D1, D2 и D2+1), не используются в других командах. Если другие команды оказывают влияние на содержание этих слов, таймер не может функционировать надлежащим образом.

**Пример**

В следующем ниже примере, когда ввод таймера CIO 000000 переходит в состояние ON, текущее значение таймера (в словах D00101 и D00100) переустанавливается в заданное значение (в словах D00101 и D00100) и начинает уменьшаться. Флаг завершения (CIO 020000) переводится в состояние ON, когда текущее значение достигает значения 00000000.

Когда CIO 000000 переходит в состояние OFF, текущее значение переустанавливается в заданное значение, а флаг завершения переводится в состояние OFF.

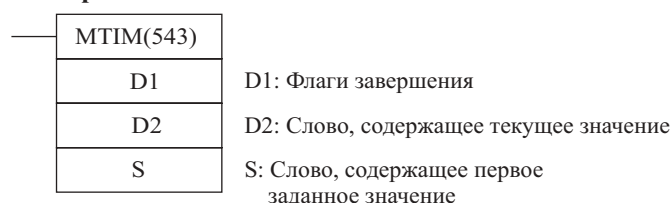


**3-6-6 Команда запуска таймера с несколькими выводами MULTI-OUTPUT TIMER MTIM(543)**

**Назначение**

Команда MTIM(543) запускает таймер с приращением текущего значения в единицах, равных 0.1 сек. Таймер имеет восемь независимых заданных значений и восемь флагов завершения. Диапазон значений от 0 до 999.9 сек. Точность отсчета таймера – от 0 до 0.01 сек.

**Символ релейно-контактной схемы**





**Модификации**

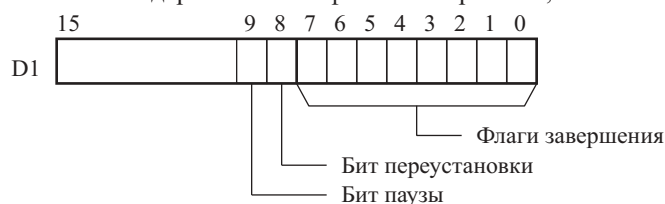
Модификации	Выполняется в каждом цикле для состояния ON.	MTIM(543)
	Выполняется один раз для дифференцирования вверх	Не поддерживается.
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не используется	Да	Да	Не используется

**Операнды****D1: Флаги завершения**

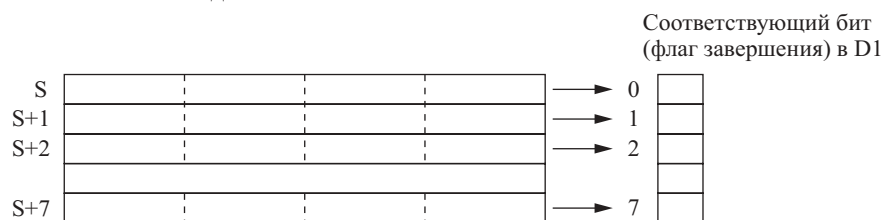
Слово D1 содержит восемь флагов завершения, а также биты паузы и переустановки.

**D2: Слово, содержащее текущее значение**

D2 содержит четырехзначное число в двоичном или двоично-десятичном коде, равное текущему значению. Диапазон изменения текущего значения от 0000 до 9999.

**S: слово, содержащее первое заданное значение**

Слова от S до S+7 содержат восемь независимых заданных значений. Диапазон установки заданного значения от 0000 до 9999.



**Примечание:** Слова S...S+7 должны находиться в одной области данных.

**Спецификации операндов**

Область	D1	D2	S
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		CIO 0000...CIO 6136
Рабочая область	W000...W511		W000...W504
Область удержания бита (Holding bit area)	H000...H511		H000...H504
Область вспомогательных битов	A448...A959		A000...A952
Область таймера	T0000...T4095		T0000...T4088
Область счетчика	C0000...C4095		C0000...C4088
Область DM	D00000...D32767		D00000...D32760
Область EM, не содержащая банков	E00000...E32767		E00000...E32760
Область EM, содержащая банки	En_00000...En_3276 (n = 0...C)		En_00000...En_32766 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	-		
Регистры данных	-	DR0...DR15	-
Индексные регистры	-		

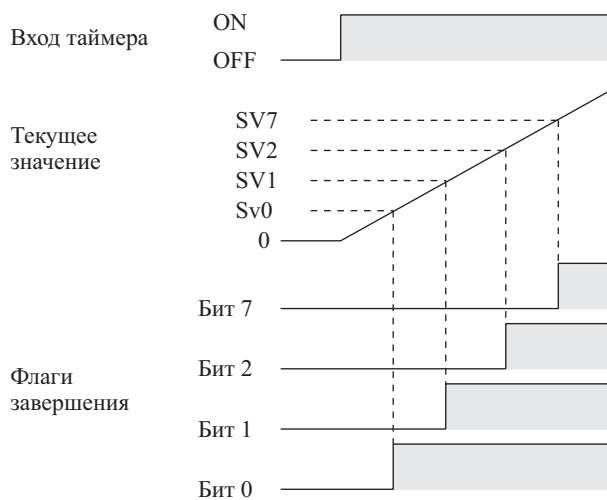
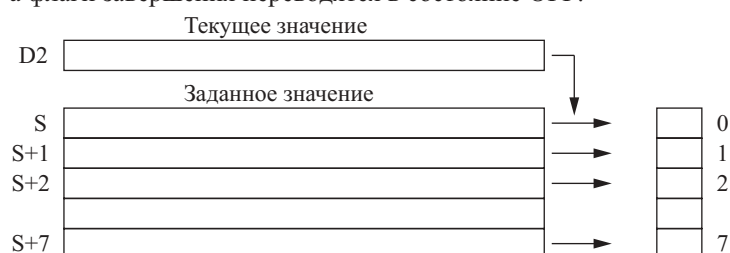
Область	D1	D2	S
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047, IR0 -2048... +2047, IR15 DR0...DR15, IR0...IR15		

### Описание

Когда условие выполнения для таймера MTIM(545) находится в состоянии ON, а бит переустановки и биты таймера – в состоянии OFF, таймер выполняет нарастающий отсчет текущего значения в D2. Если бит паузы переводится в состояние ON, выполнение отсчета прерывается, однако текущее значение сохраняется. Отсчет возобновляется, когда бит паузы переводится в состояние OFF.

Текущее значение (содержание D2) сравнивается с восемью заданными значениями, содержащимися в словах от S до S+7 при каждом выполнении команды MTIM(545) и если одно из заданных значений меньше или равно текущему значению, соответствующий флаг завершения (слово D1 биты от 00 до 07) переводится в состояние ON.

Когда текущее значение достигает величины, равной 9999, текущее значение переустанавливается в значение 0000, а все флаги завершения переводятся в состояние OFF. Если в процессе работы таймера или в паузе бит переустановки переключается в состояние ON, текущее значение сбрасывается в значение 0000, а флаги завершения переводятся в состояние OFF.



В следующей ниже таблице представлена работа таймера MTIM(543) при возможных четырех комбинациях состояний бита переустановки и бита паузы.

Бит переустановки	Бит паузы	Функционирование
OFF	OFF	Текущее значение обновляется и соответствующий флаг завершения переводится в состояние ON, когда заданное значение меньше или равно текущему значению.
	ON	Текущее значение не обновляется и команда MTIM(543) выполняется как команда NOP(000).
ON	OFF	Текущее значение сбрасывается в значение 0000, и флаг завершения переводится в состояние OFF. Текущее значение не обновляется.
	ON	

Бит переустановки и бит паузы действуют только в том случае, когда условие выполнения команды MTIM(543) находится в состоянии ON.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда текущее значение в слове D2 выражено не в двоично-десятичном коде. Переводится в состояние OFF в других случаях.

**Меры предосторожности**

В отличие от других таймеров, таймер МТІМ(543) не использует номера. (Для таймера МТІМ(543) операция регенерации текущего значения области таймера не выполняется.)

Когда текущее значение достигает величины, равной 9999, текущее значение переустанавливается в значение 0000, а все флаги завершения переводятся в состояние OFF.

Если заданное значение в словах S S+7 содержат данные, выраженные не в двоично-десятичном коде, эти данные игнорируются. В этом случае ошибка не определяется и флаг завершения своего состояния не изменяет.

Так как флаг завершения таймера МТІМ(543) находится в области данных, его состояние может принудительно изменяться, подобно другим битам, однако при этом текущее значение не изменяется.

Когда в процессе работы требуется использование восьми или менее восьми заданных значений, слово, следующее вслед за последним заданным значением, установите в значение, равное 0000. При выполнении команды МТІМ(543) заданное значение, величина которого равна 0000, игнорируется. Кроме того, все последующие заданные величины также не используются.

SCH	0002CH	1	0	2	9	↕ Эти заданные значения игнорируются
	0003CH	2	5	0	6	
	0004CH	6	0	4	0	
	0005CH	0	0	0	0	
S+7CH	0009CH					

Регенерация текущего значения таймера осуществляется только при выполнении команды МТІМ(543). Поэтому таймер не будет работать надлежащим образом, если длительность цикла превышает 100 мсек., т.к. единица приращения таймера равна 100 мсек. В целях обеспечения нормального функционирования таймера и предотвращения проблем, связанных с большой длительностью цикла, вводите одну и ту же команду МТІМ(543) в нескольких точках программы.

Регенерация состояния флага завершения производится только при выполнении команды МТІМ(543), поэтому для перевода флага в состояние ON после завершения отсчета может потребоваться задержка, длительностью до одного цикла.

Когда команда МТІМ(543) располагается в части программы, находящейся между командами ІL(003) и ІLC(003), и программа блокирована, текущее значение удерживается (не сбрасывается). При расположении команды МТІМ(543) между указанными выше командами, принимайте этот факт во внимание.

Когда таймер МТІМ(543) расположен в части программы, находящейся между командами JMP(004) и JME(005), и часть программы обходится, текущее значение сохраняется. При расположении таймера МТІМ(543) между командами JMP(004) и JME(005), принимайте этот факт во внимание.

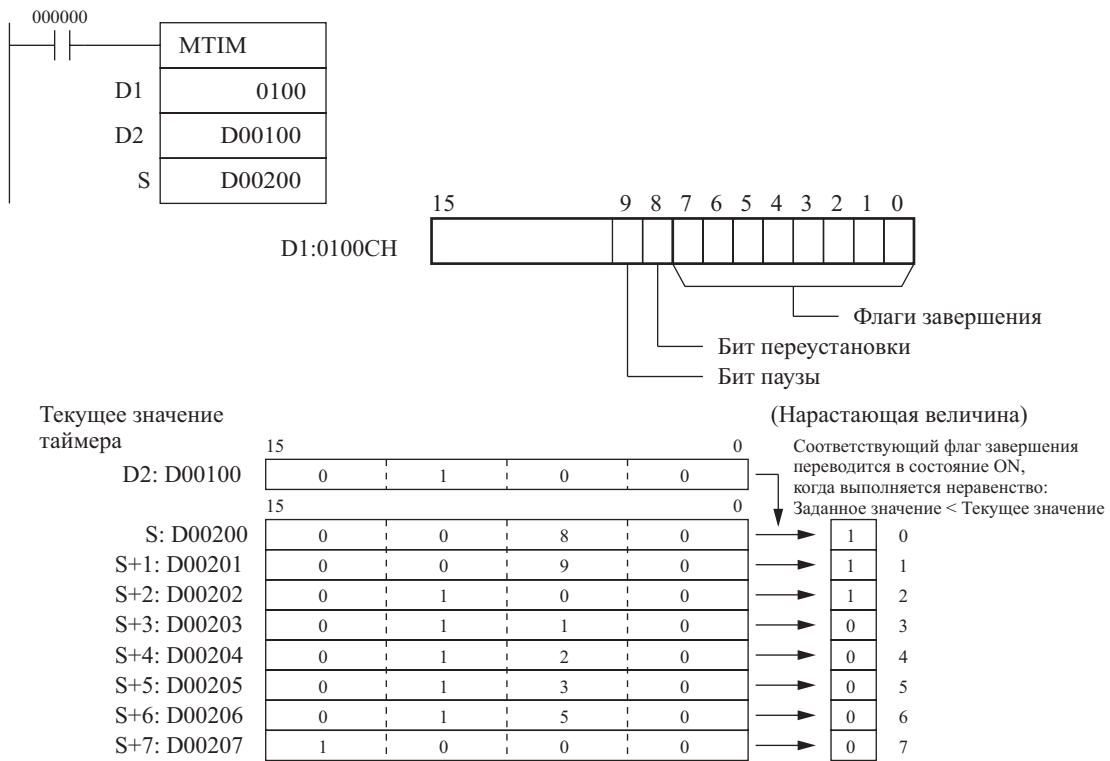
Убедитесь в том, что слова, указанные для флагов завершения и текущего значения (D1, D2), не используются в других командах. Если другие команды оказывают влияние на содержание этих слов, таймер не может функционировать надлежащим образом.

Если для D1 указывается слово в области CIO, команды SET и RSET могут использоваться для управления битом паузы и битом переустановки.

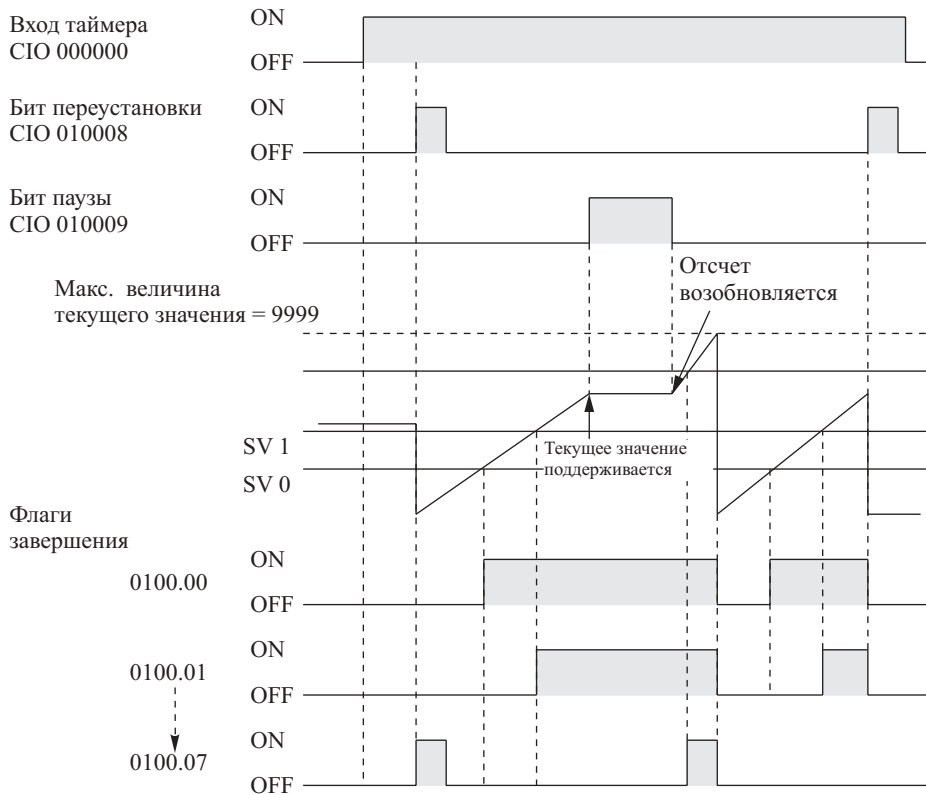
**Пример**

В следующем ниже примере, если CIO 000000 находится в состоянии ON, а бит паузы (CIO 010009) – в состоянии OFF, таймер начинает отсчет, когда бит переустановки (CIO 010009) переводится из состояния ON в состояние OFF. Нарастающий отсчет текущего значения начинается со значения, равного 0000.

Восемь заданных значений, содержащихся в словах D00200 D00207, сравниваются с текущим значением, и когда текущее значение достигает заданной величины или превышает заданную величину, соответствующий флаг завершения (CIO 010000 CIO 010007) переводится в состояние ON.



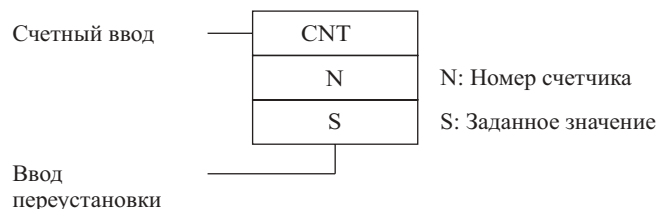
В процессе выполнения отсчета вход должен оставаться в состоянии ON



## 3-6-7 Команда запуска счетчика COUNTER: CNT

**Назначение**

Команда CNT запускает счетчик с отрицательным приращением. Пределы установки заданного значения счетчика – от 0 до 9999.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	CNT
	Выполняется один раз для дифференцирования вверх	Не поддерживается.
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не используется	Да	Да	Да

**Операнды****N: Номер счетчика**

Номер счетчика должен находиться в пределах от 0000 до 4095.

**S: заданное значение**

Заданное значение должно находиться в пределах от 0000 до 9999.

**Спецификации операндов**

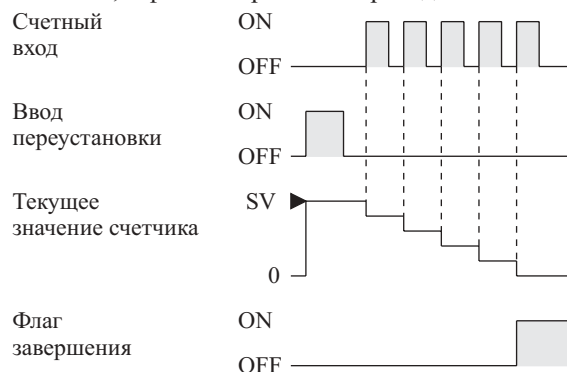
Область	N	S
Область ввода/вывода (область CIO)	–	CIO 0000...CIO 6143
Рабочая область	–	W000...W511
Область удержания бита (Holding bit area)	–	H000...H511
Область вспомогательных битов	–	A000...A959
Область таймера	–	T0000...T4095
Область счетчика	C0000...C4095	C0000...C4095
Область DM	–	D00000...D32767
Область EM, не содержащая банков	–	E00000...E32767
Область EM, содержащая банки	–	En_00000...En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	–	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)
Константы	–	0000...9999
Регистры данных	–	DR0...DR15
Индексные регистры	–	...
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15	

**Описание**

Текущее значение счетчика уменьшается на единицу каждый раз, когда счетный ввод переводится из состояния OFF в состояние ON. Флаг завершения переводится в состояние ON. Когда текущее значение достигает величины, равной 0.

После того, как Флаг завершения переводится в состояние ON, производите переустановку счетчика посредством перевода ввода переустановки в состояние ON или с помощью команды CNR(545). В противном случае счетчик повторно не запускается.

Когда на ввод переустановки подается высокий уровень (ON), счетчик переустанавливается, и состояние на счетном входе игнорируется. (При переустановке счетчика текущее значение изменяется на заданное значение, а флаг завершения переводится в состояние OFF.)



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	<p>Переводится в состояние ON, когда адрес N косвенно указывается через индексный регистр, однако адрес в индексном регистре не является адресом текущего значения счетчика.</p> <p>Переводится в состояние ON, если S не содержит данных в двоично-десятичном коде.</p> <p>Переводится в состояние OFF в других случаях.</p>

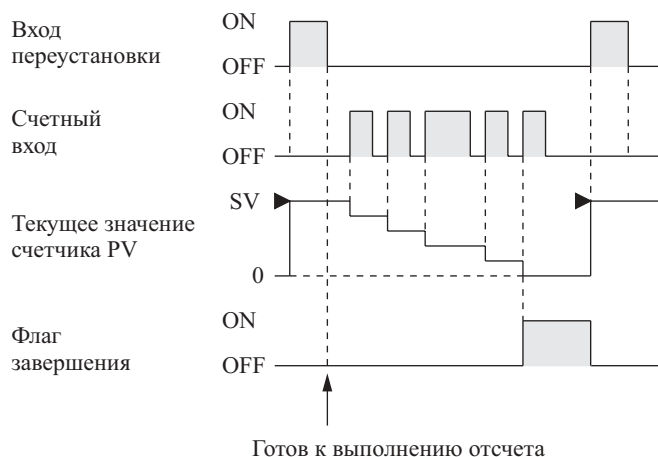
### Меры предосторожности

Номера счетчиков распределяются между командами CNT, CNTR(012), и CNTW(814). Если один и тот же номер используется двумя счетчиками, однако, счетчики не используются одновременно, при проверке программы определяется ошибка дублирования номера, тем не менее, оба счетчика функционируют нормально. Счетчики, использующие один и тот же номер, не могут функционировать надлежащим образом при одновременной работе.

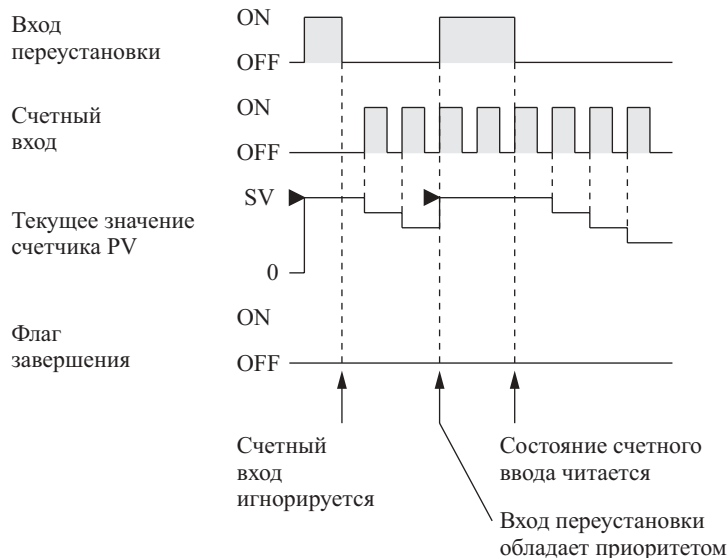
Текущее значение счетчика подвергается регенерации, когда счетный вход переходит из состояния OFF в состояние ON, а состояние флага завершения обновляется при каждом выполнении команды CNT. Флаг завершения переводится в состояние ON, когда текущее значение равно 0, и переводится в состояние OFF, когда текущее значение отлично от 0.

В случае, когда производится принудительная установка счетчика CNT, соответствующий флаг завершения переводится в состояние ON, а текущее значение счетчика устанавливается в значение 0000. Когда производится принудительный сброс счетчика CNT, его флаг завершения переводится в состояние OFF, а текущее значение изменяется на заданное значение.

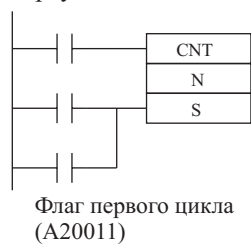
Непрерывно переустанавливайте счетчик путем изменения состояния ввода переустановки по схеме OFF ON OFF перед началом отсчета через счетный вход, как показано на следующем ниже рисунке. Если на вводе переустановки поддерживается состояние ON, состояние счетного входа игнорируется.



Вход переустановки обладает приоритетом относительно счетного входа, поэтому, когда на обоих входах одновременно появляется состояние ON, счетчик переустанавливается. (Текущее значение заменяется заданным значением, а флаг завершения переводится в состояние OFF).



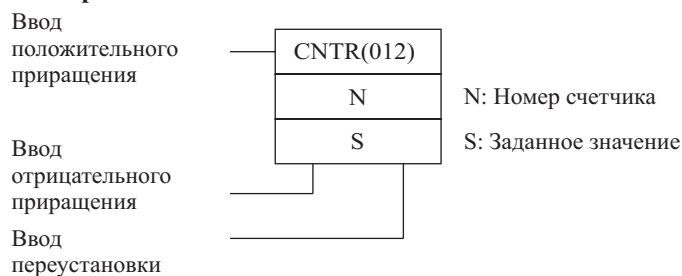
Текущее значение сохраняется даже при прерывании подачи питания. Если вам необходимо возобновить работу счетчика с заданного значения, вместо сохраненного текущего значения, вводите в качестве входа переустановки счетчика флаг первого цикла (A20011).



### 3-6-8 Команда запуска реверсивного счетчика REVERSIBLE COUNTER: CNTR(012)

#### Назначение

Команда CNT запускает реверсивный счетчик.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	CNTR(012)
	Выполняется один раз для дифференцирования вверх.	Не поддерживается.
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не используется	Да	Да	Да

**Операнды****N: Номер счетчика**

Номер счетчика должен находиться в пределах от 0000 до 4095.

**S: заданное значение**

Заданное значение должно находиться в пределах от 0000 до 9999.

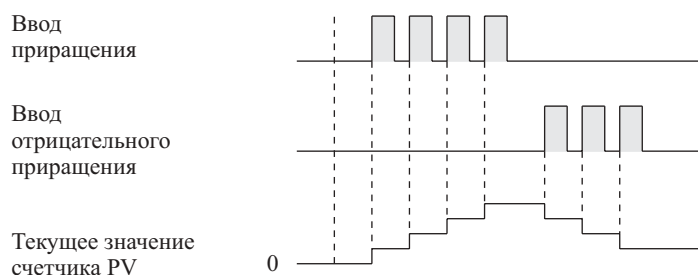
**Спецификации операндов**

Область	N	S
Область ввода/вывода (область CIO)	—	CIO 0000...CIO 6143
Рабочая область	—	W000...W511
Область удержания бита (Holding bit area)	—	H000...H511
Область вспомогательных битов	—	A000...A959
Область таймера	—	T0000...T4095
Область счетчика	C0000...C4095	C0000...C4095
Область DM	—	D00000юD32767
Область EM, не содержащая банков	—	E00000...E32767
Область EM, содержащая банки	—	En_0000...En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	-	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)
Константы	—	0000...9999
Регистры данных	—	DR0...DR15
Индексные регистры	-	...
Косвенная адресация с использованием индексных регистров	,IR0...,IR15, -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15	

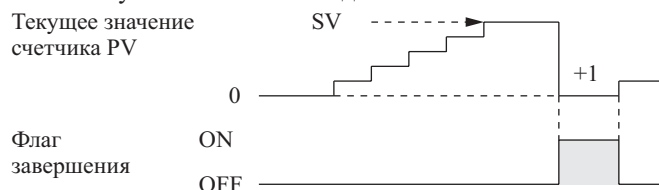
**Описание**

Текущее значение счетчика увеличивается на единицу каждый раз, когда ввод приращения переводится из состояния OFF в состояние ON. Текущее значение уменьшается, когда аналогичное происходит на вводе отрицательного приращения. Текущее значение может изменяться в различном направлении в диапазоне от 0 до заданного значения.





При нарастающем отсчете флаг завершения переводится в состояние ON, когда после увеличения до заданного значения текущее значение сбрасывается в 0. Флаг переводится в состояние OFF, когда текущее значение увеличивается от 0 до 1.



При убывающем (обратном) отсчете флаг завершения переводится в состояние ON, когда после завершения отсчета, текущее значение из нуля переводится в заданное значение. Флаг переводится в состояние OFF, когда текущее значение уменьшается от заданного значения на единицу.



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда адрес N косвенно указывается через индексный регистр, однако адрес в индексном регистре не является адресом текущего значения счетчика. Переводится в состояние ON, если S не содержит данных в двоично-десятичном коде. Переводится в состояние OFF в других случаях.

### Меры предосторожности

Номера счетчиков распределяются между командами CNT, CNTR(012), и CNTW(814). Если один и тот же номер используется двумя счетчиками, однако счетчики не используются одновременно, при проверке программы определяется ошибка дублирования номера, тем не менее, оба счетчика функционируют нормально. Счетчики, использующие один и тот же номер, не могут функционировать надлежащим образом при одновременной работе.

В случае, когда состояние обоих вводов одновременно переводится из состояния OFF в состояние ON, текущее значение счетчика не изменяется. Когда на вводе переустановки присутствует состояние ON, текущее значение сбрасывается в 0, а состояние обоих счетных вводов игнорируется.

Флаг завершения находится в состоянии ON, только тогда, когда после увеличения текущего значения от 0 до заданного значения, текущее значение сбрасывается в 0, или когда после уменьшения текущего значения до 0, оно изменяется на заданное значение. Во всех прочих случаях флаг завершения находится в состоянии OFF.

При вводе команды CNTR(012) в мнемоническом виде, вначале вводите ввод приращения (II), затем ввод отрицательного приращения (DI), затем вход переустановки (R), и, наконец, команду CNTR(012). При составлении релейно-контактной схемы вначале вводите ввод приращения (II), затем команду CNTR(012), затем ввод отрицательного приращения (DI), затем вход переустановки (R).

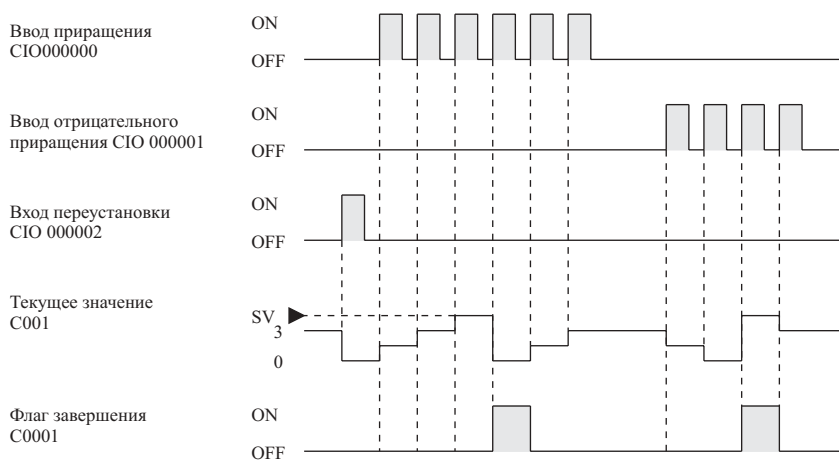
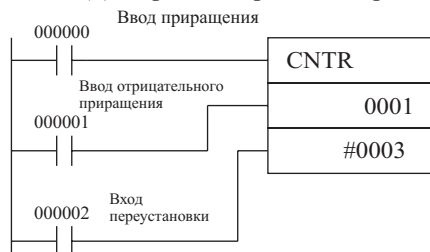
### Примеры

#### Основной принцип работы команды CNTR(012)

Текущее значение счетчика сбрасывается в 0 посредством перевода ввода переустановки (CIO000002) вначале в состояние ON, а затем в состояние OFF. Текущее значение увеличивается на единицу каждый

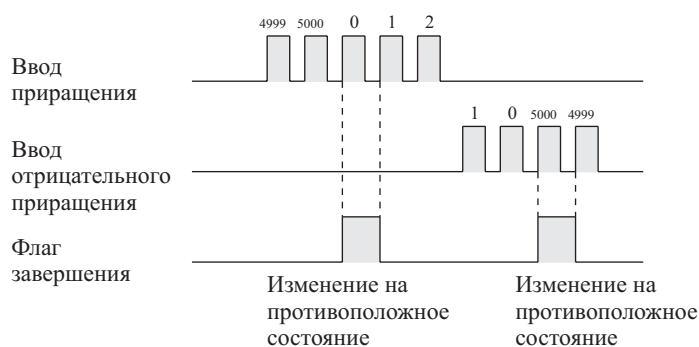
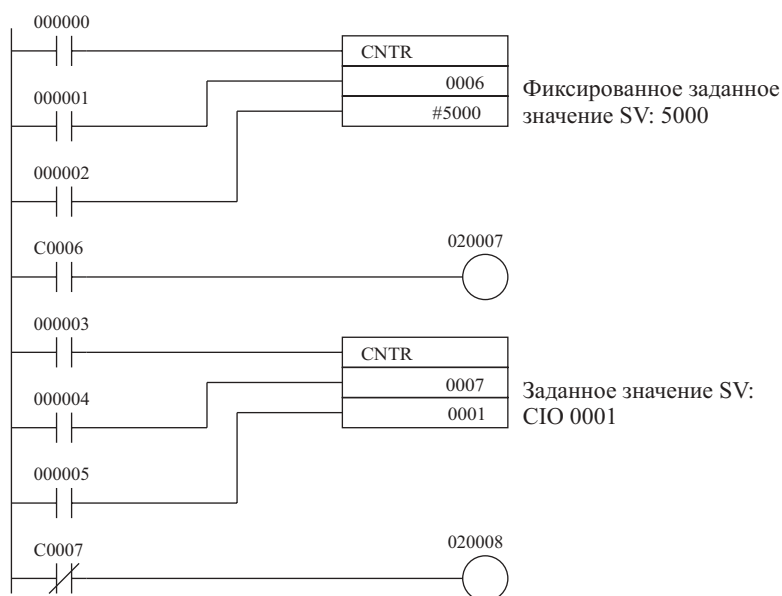
раз, когда ввод приращения (CIO 000000) переходит из состояния OFF в состояние ON. Когда текущее значение увеличивается после достижения заданного значения (3), оно автоматически переустанавливается в 0 и флаг завершения переводится в состояние ON.

Подобно изложенному выше, текущее значение уменьшается на единицу каждый раз, когда ввод отрицательного приращения (CIO000001) переходит из состояния OFF в состояние ON. Когда текущее значение уменьшается на единицу после достижения нуля, оно автоматически переустанавливается в заданное значение (3), и флаг завершения переводится в состояние ON.



#### Указание заданного значения в слове

В следующем ниже примере заданное значение для CNTR(012) 0007 определяется содержанием CIO 0001.

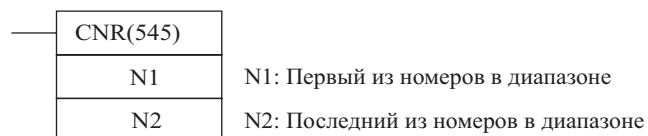


### 3-6-9 Команда выполнения переустановки таймера/счетчика RESET TIMER/COUNTER: CNR(545)

#### Назначение

Команда CNR(545) осуществляет переустановку таймеров или счетчиков в указанном диапазоне их номеров.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	CNTR(545)
	Выполняется один раз для дифференцирования вверх	@CNTR(545).
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

##### N1: Первый из номеров в диапазоне номеров

Номер N1 должен быть номером таймера от T0000 до T4095 или номером счетчика от C0000 до C4095.

**N2: последний из номеров в диапазоне номеров**

Номер N2 должен быть номером таймера от T0000 до T4095 или номером счетчика от C0000 до C4095.

**Примечание:** N1 и N2 должны находиться в одной области данных, т.е. N1 и N2 должны быть либо номерами таймеров, либо номерами счетчиков.

**Спецификации операндов**

Область	N1	N2
Область ввода/вывода (область CIO)	–	–
Рабочая область	–	–
Область удержания бита (Holding bit area)	–	–
Область вспомогательных битов	–	–
Область таймера	T0000...T4095	T0000...T4095
Область счетчика	C0000...C4095	C0000...C4095
Область DM	–	–
Область EM, не содержащая банков	–	–
Область EM, содержащая банки	–	–
Адреса DM/EM памяти при косвенном указании в двоичном коде	–	–
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–	–
Константы	–	–
Регистры данных	–	–
Индексные регистры	–	–
Косвенная адресация с использованием индексных регистров	,IR0...,IR15, -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-)IR15	

**Описание**

По команде CNR(545) производится переустановка флагов завершения всех таймеров и счетчиков от N1 до N2. В то же время текущие значения устанавливаются в максимальные значения. (Текущее значение заменяется заданным значением после выполнения следующей команды управления таймером или счетчиком.)

**Переустановка таймеров по команде CNR(545)**

Осуществляется переустановка (сброс) значений таймеров TIM, TIMH(015), TMHH(540), TTIM(087), TIMW(813) и TMHW(815) в указанном диапазоне их номеров. При переустановке таймера, соответствующий флаг завершения переводится в состояние OFF, а текущее значение устанавливается в максимальное значение, равное 9999.

**Примечание:** Таймеры TIML(542) и MTIM(543) по команде CNR(545) не переустанавливаются, т.к. этим таймерам не присваиваются номера.

**Переустановка счетчиков по команде CNR(545)**

Осуществляется переустановка (сброс) значений счетчиков CNT, CNTR(012), и CNTW(814) в указанном диапазоне их номеров. При переустановке счетчика, соответствующий флаг завершения переводится в состояние OFF, а текущее значение устанавливается в максимальное значение, равное 9999.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда номер N1 косвенно указывается через индексный регистр, однако адрес в индексном регистре не является адресом текущего значения таймера или счетчика. Переводится в состояние ON, когда номер N2 косвенно указывается через индексный регистр, однако адрес в индексном регистре не является адресом текущего значения таймера или счетчика. Переводится в состояние ON, если номера N1 и N2 находятся в различных областях данных. Переводится в состояние OFF в других случаях.

**Меры предосторожности**

Команда CNR(545) не производит переустановку собственно команд управления таймерами и счетчиками, она производит переустановку текущих значений и флагов завершения, присвоенных данным коман-

дам. В большинстве случаев, результат выполнения команды CNR(545) отличается от переустановки команд.

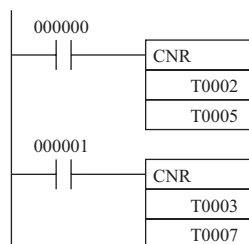
Например, когда команда TIM переустанавливается, текущее значение заменяется заданным значением. Однако когда таймер переустанавливается по команде CNR(545), его текущее значение устанавливается в максимальное значение, равное 9999.

В случае, когда задаваемый номер  $N1 > N2$ , осуществляется переустановка только флага завершения.

#### Пример

В следующем примере, когда СЮ 000000 находится в состоянии ON, флаги завершения таймеров T0002 и T0005 переводятся в состояние OFF, а текущие значения устанавливаются в максимальное значение, равное 9999.

Когда СЮ 000001 находится в состоянии ON, флаги завершения счетчиков C0003 и C0007 переводятся в состояние OFF, а текущие значения устанавливаются в максимальное значение, равное 9999.



#### 3-6-10 Пример использования таймера и счетчика

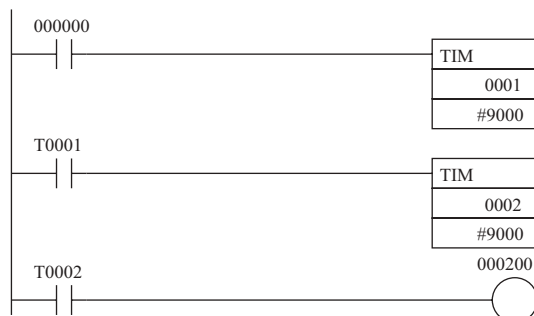
Ниже представлены различные примеры применения команд управления таймером и счетчиком, включая долговременные таймеры, двухступенчатый таймер, использование задержки при включении/выключении, одноктактный бит, и мерцающего бита.

#### Пример 1: Долговременные таймеры

В следующем примере программы показано три пути создания долговременных таймеров при помощи стандартных команд TIM и CNT.

##### Две команды TIM

В данном примере при помощи комбинации из двух команд TIM создается один 30-ти минутный таймер.



Адрес	Команда	Операнды
000000	LD	000000
000001	TIM	0001
		#9000
000002	LD	T 00001
000003	TIM	0002
		#9000
000004	LD	T 0002
000005	OUT	000200

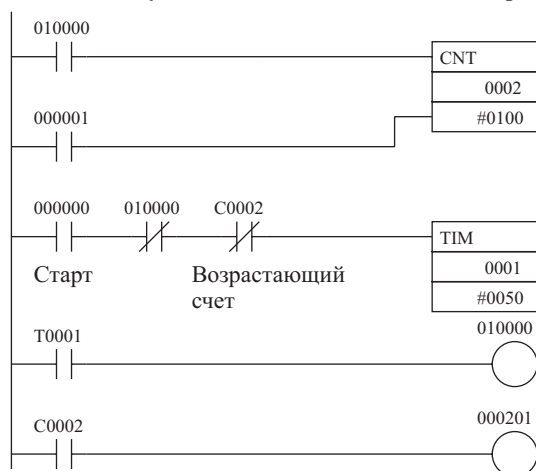
##### Команды TIM и CNT

В данном примере при помощи команды TIM и CNT создается 500 – секундный таймер.

TIM 0001 генерирует один импульс каждые 5 сек., а CNT 0002 выполняет подсчет этих импульсов. Для такой комбинации команд заданным значением является величина, равная интервалу таймера заданное значение счетчика. В данном случае заданное значение определяется следующим образом:

$$5 \text{ сек} \times 100 = 500 \text{ сек.}$$

С помощью такой комбинации команд текущее значение долговременного таймера в действительности является текущим значением счетчика, которое поддерживается в процессе прерываний подачи питания.

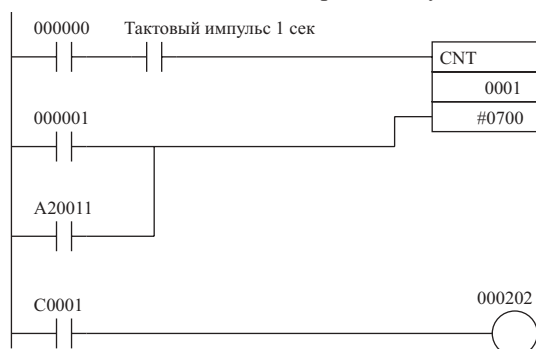


Адрес	Команда	Операнды
000000	LD	010000
000001	LD	000001
000002	CNT	0002 #0100
000003	LD	000000
000004	AND NOT	010000
000005	AND NOT	C 0002
000006	TIM	0001 #0050
000007	LD	T 0001
000008	OUT	010000
000009	LD	C 0002
000010	OUT	000201

#### Тактовый импульс и команда CNT

В данном примере для получения 700 – секундного таймера по команде CNT производится подсчет секундных тактовых импульсов.

Если флаг первого цикла (A20011) используется в качестве дополнения для ввода переустановки счетчика, при запуске программы текущее значение счетчика будет сбрасываться в заданное значение, вместо того, чтобы использовать предшествующее текущее значение.

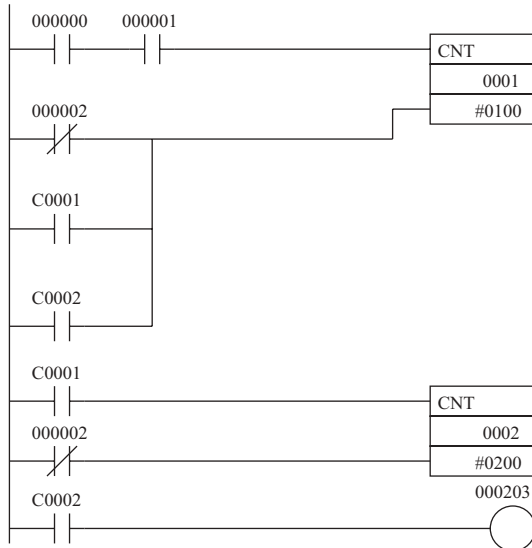


Адрес	Команда	Операнды
000000	LD	0000000
000001	AND	1 сек.
000002	LD NOT	000001

Адрес	Команда	Операнды
000003	CNT	0001 #0700
000004	LD	C 0001
000005	OUT	000202

**Пример 2: Двухступенчатый таймер**

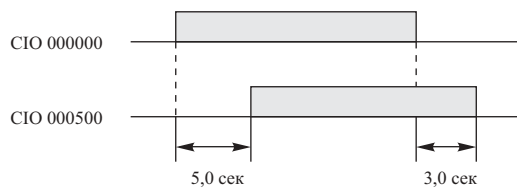
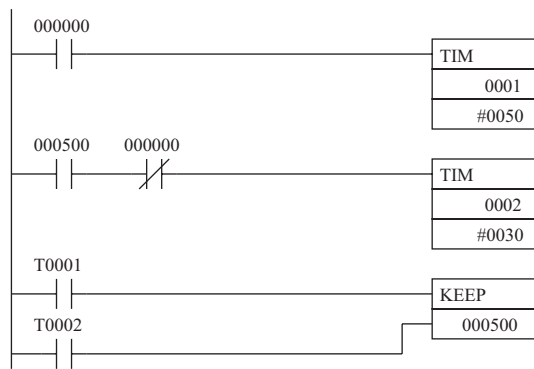
Когда требуемое заданное значение превышает число 65535, для выполнения такой задачи можно объединять два счетчика, как показано в следующем ниже примере. В данном случае, две команды CNT объединены для создания счетчика двоично-десятичных чисел, с заданным значением, равным 20000.



Адрес	Команда	Операнды
000000	LD	000000
000001	AND	000001
000002	LD NOT	000002
000003	OR	C 0001
000004	OR	C 0002
000005	CNT	0001 #0100
000006	LD	C 0001
000007	LD NOT	000002
000008	CNT	0002 #0200
000009	LD	C 0002
000010	OUT	000203

**Пример 3: Задержка включения/выключения (ON/OFF)**

В данном примере два таймера TIM объединены с командой KEEP(011) для осуществления задержки перевода в состояние ON и задержки перевода в состояние OFF. CIO 000500 будет переводиться в состояние ON через 5 секунд после перехода CIO 000000 в состояние ON. Переход в состояние OFF происходит спустя 3 секунды после перехода CIO 000000 в состояние OFF.

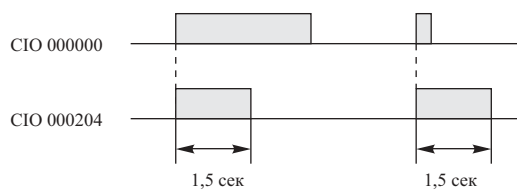
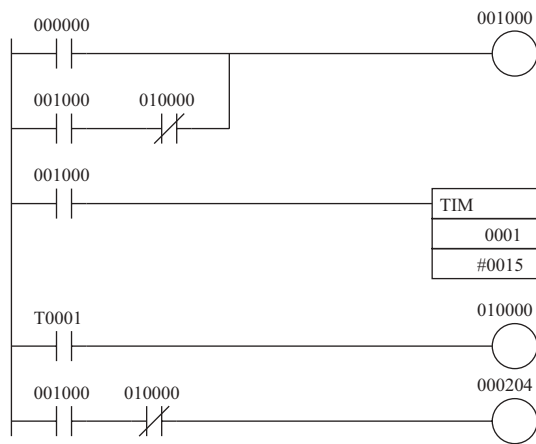


Адрес	Команда	Операнды
000000	LD	000000
000001	TIM	0001
		#0050
000002	LD	000500
000003	AND NOT	000000
000004	TIM	0002
		#0030
000005	LD	T 0001
000006	LD	T 0002
000007	KEEP(011)	000500

**Пример 4: Однотактный бит**

Команда TIM может объединяться с командами OUT и OUT NOT для контроля длительности пребывания определенного бита в состоянии ON или в состоянии OFF. В данном примере, CIO 000204 будет находиться в состоянии ON в течение 1.5 секунд (заданное значение для T0001), после того, как CIO 000000 переводится в состояние ON.





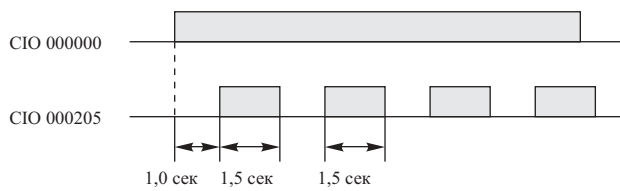
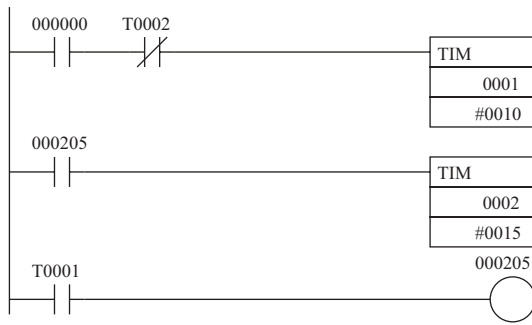
Адрес	Команда	Операнды
000000	LD	000000
000001	LD	001000
000002	AND NOT	010000
000003	OR	000000
000004	OUT	001000
000005	LD	001000
000006	TIM	0001 #0015
000007	LD	T 0001
000008	OUT	010000
000009	LD	001000
000010	AND NOT	010000
000011	OUT	000204

**Пример 5: Мерцающий бит**

Следующий ниже пример программы показывает два способа создания мерцающих битов. Второй пример является только имитацией тактового импульса.

**Две команды TIM**

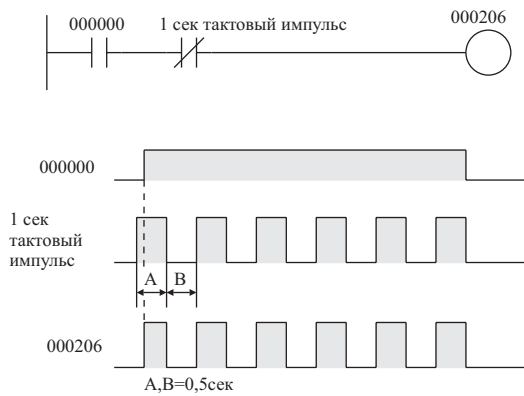
Две команды TIM могут объединяться для принудительного перевода бита в состояние ON и OFF через установленный интервал времени, когда условие выполнения находится в состоянии ON. В данном примере, CIO000205 будет находиться в состоянии OFF в течение 1 секунды, затем в состоянии ON в течение 1.5 секунд, до тех пор, пока CIO 000000 будет находиться в состоянии ON.



Адрес	Команда	Операнды
000000	LD	000000
000001	AND	T 0002
000002	TIM	0001 #0010
000003	LD	000205
000004	TIM	0002 #0015
000005	LD	T 0001
000006	OUT	000205

### Тактовый импульс

Требуемое условие выполнения может объединяться с тактовым импульсом, для имитации тактового импульса (0.1 сек. 0.2 сек. 1.0 сек.).



Адрес	Команда	Операнды
000000	LD	000000
000001	AND	1 сек.
000002	OUT	000206

### 3-6-11 Косвенное указание номеров таймеров/счетчиков

Косвенное указание номеров таймеров и счетчиков может производиться с помощью индексных регистров. При использовании индексных регистров, для указания в желаемом индексном регистре адреса теку-

шего значения таймера или счетчика применяйте команду MOVRW(561)(MOVE TIMER/COUNTER PV TO REGISTER).

С помощью индексных регистров может производиться косвенная адресация следующих таймеров и счетчиков: TIM, TIMH(015), TTIM(087), TMH(540), TIMW(813), TMHW(815), CNT, CNTR(012), CNTW(814). (Перечисленным таймерам и счетчикам присваиваются номера.)

Команды управления таймерами и счетчиками выполняться не будут, если адрес памяти внутреннего обмена в указанном индексном регистре не является адресом текущего значения таймера или счетчика.

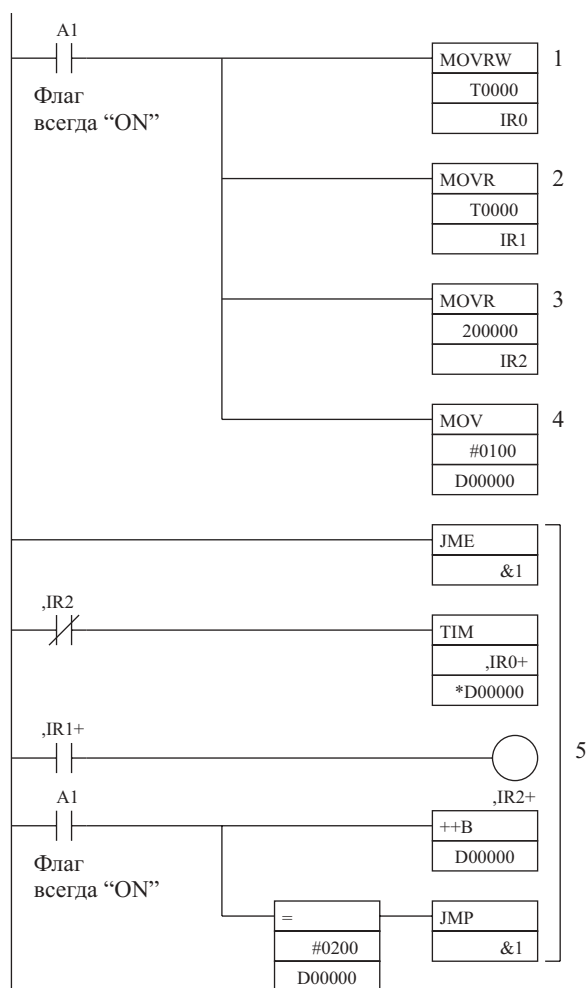
Использование индексных регистров для косвенной адресации таймеров и счетчиков может способствовать сокращению объема программы и увеличить ее гибкость. Например, в этом случае можно создавать подпрограммы.

### Пример

В данном примере показан раздел программы, использующий косвенную адресацию для определения и запуска 100 таймеров с заданными значениями, содержащимися в D00100 D00199.

Регистр IR0 содержит адрес памяти внутреннего ввода/вывода, принадлежащий текущему значению таймера, а регистр IR1 содержит адрес памяти внутреннего ввода/вывода, принадлежащий флагу завершения таймера.

Адрес DM памяти	Содержание	Функция
D00100	0010	Заданное значение для T0000
D00101	0100	Заданное значение для T0001
D00102	0050	Заданное значение для T0002
–	–	–
D00199	0999	Заданное значение для T0099



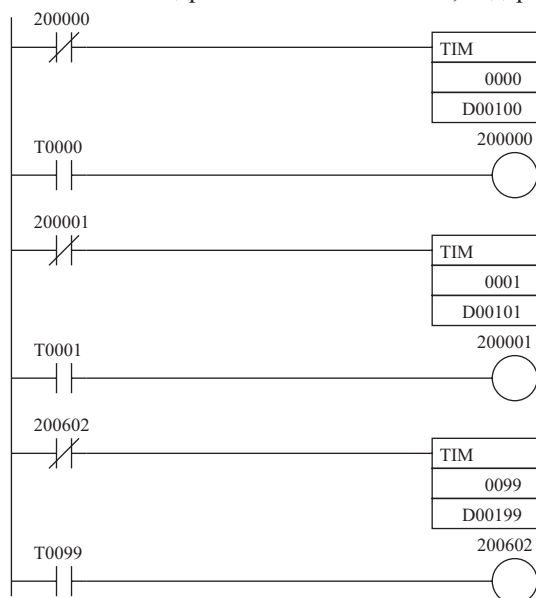
- 1,2,3...**
1. Команда *MOVRW*(561) перемещает адрес памяти внутреннего ввода/вывода, принадлежащий текущему значению таймера *T0000*, в регистр *T0*. Впоследствии регистр *IR0* может использоваться вместо номера таймера.
  2. Команда *MOVR*(560) перемещает адрес памяти внутреннего ввода/вывода, принадлежащий флагу завершения таймера *T0000*, в регистр *IR1*.
  3. Команда *MOVR*(560) перемещает адрес памяти внутреннего ввода/вывода, принадлежащий *CIO 200000*, в регистр *IR2*.
  4. Команда *MOV*(021) перемещает *#0100* в *D00000* при косвенной адресации заданного значения таймера.
  5. Содержание *IR0*, *IR1*, *IR2* и *D00000* увеличивается на единицу каждый раз, когда данный цикл выполняется 100 раз, запуская таймеры *T0000* *T0099*.

**Внимание!** Цикл *JMP*(004) – *JME*(005) повторяется до тех пор, пока содержание *D00000* не равно *#200*. После ста повторений в *D00000* будет содержаться *#0200* и выполнение цикла заканчивается.

**Опасность!** Не применяйте переход с номером *0000* в приведенном выше типе программ.

В приведенной выше программе цикл содержит 4 входных параметра, используемых для запуска 100 таймеров при помощи обычной подпрограммы.

- *IR0* Адрес памяти внутреннего ввода/вывода, содержащий текущее значение таймера.
- *IR1* Адрес памяти внутреннего ввода/вывода, содержащий флаг завершения таймера.
- *IR2* Адрес памяти внутреннего ввода/вывода, используемый условием выполнения.
- *D00000* Адрес слова в памяти *DM*, содержащий заданное значение таймера.



Адрес	Команда	Операнды
000000	LD NOT	200000
000001	TIM	0000 D 00100
000002	LD	T 0000
000003	OUT	200000
000004	LD NOT	200001
000005	TIM	0001 D 00101
000006	LD	T 0001
000007	OUT	200001
000008	LD NOT	200002
000009	TIM	0002 D 00102
000010	LD	T 0002

3-6 Команды управления таймерами и счетчиками

---

Адрес	Команда	Операнды
000011	OUT	200002
...		
000396	LD NOT	200602
000397	TIM	0099
		D 00199
000398	LD	T 0000
000399	OUT	200602

## 3-7 Команды сравнения

В настоящем разделе приводится описание команд, используемых для сравнения данных различной длины, и применение различных способов сравнения.

Команда	Мнемоническое изображение	Функциональный код	Страница
Команды сравнения	LD, AND, OR =, <>, <, <=, >, >=, L, S	300...328	175
COMPARE	CMP	020	180
DOUBLE COMPARE	CMPL	060	182
SIGNED BINARY COMPARE	CPS	114	185
DOUBLE SIGNED BINARY COMPARE	CPSL	115	187
MULTIPLE COMPARE	MCMP	019	190
TABLE COMPARE	TCMP	085	192
BLOCK COMPARE	BCMP	068	195

## 3-7-1 Команды сравнения ввода (300...328)

**Назначение**

Команды сравнения ввода осуществляют сравнение двух величин (констант и/или содержания указанных слов) и создают состояние выполнения ON, когда условие сравнения является истиной. Команды сравнения ввода могут использоваться для сравнения данных со знаком или данных без знака длиной в одно слово или длиной в два слова.

**Символы релейно-контактной схемы**

Символ & возможности	
S1	S1: Сравнимые данные 1
S2	S2: Сравнимые данные 2

**Модификации**

Модификации	Создает состояние ON в каждом из циклов, в котором результат сравнения верен.	Команды сравнения
Указание немедленной регенерации		Не поддерживается

**Применяемые области программы**

Области блоков программы	Области шагов программы	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов команд для данных, состоящих из одного слова**

Область	S1	S2
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A000...A959	
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#0000...#FFFF (двоичн.)	
Регистры данных	DR0...DR15	
Индексные регистры	-	

Область	S1	S2
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 , IR15 DR0...DR15, IR0...IR15 ,IR0+(++)... ,IR15+(++) ,-(-)IR0...,-(-)IR15	

#### Спецификации операндов команд для данных двойной длины

Область	S1	S2
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000... *En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFFF (двоичн.)	
Регистры данных	—	
Индексные регистры	IR0...IR15 (только для данных без знака)	
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 , IR15 DR0...DR15, IR0...IR15 ,IR0+(++)... ,IR15+(++) ,-(-)IR0...,-(-)IR15	

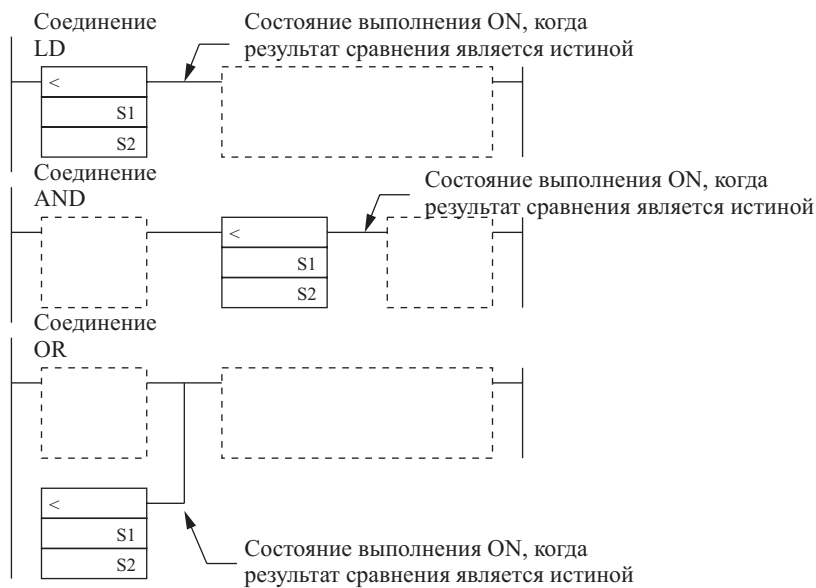
#### Описание

Команда сравнения ввода выполняет сравнение данных со знаком или без знака S1 и S2, и создает состояние выполнения ON, когда условие сравнения является истиной. В отличие от команд CMP(020) и CMPL(060) результат выполнения команды сравнения ввода выражается непосредственно в виде состояния выполнения. Вследствие изложенного, программа упрощается и становится более быстрой, т.к. отпадает необходимость осуществления доступа к результату сравнения через Арифметические флаги.

#### Ввод команд

Команды сравнения ввода используются подобно командам LD, AND, OR для управления выполнением последовательных команд.

Тип ввода	Функционирование
LD	Команда может непосредственно присоединяться к левой линии шины.
AND	Команда не может непосредственно присоединяться к левой линии шины.
OR	Команда может непосредственно присоединяться к левой линии шины.



### Свойства

Команды сравнения ввода производят сравнение данных со знаком или без знака, а также данных, состоящих из одного слова и данных двойной длины. Если свойства не указываются, осуществляется сравнение данных, состоящих из одного слова, и не имеющих знака. С учетом трех типов ввода и двух дополнительных свойств, всего существует 72 типа команд сравнения.

Символ	Дополнительная возможность (формат данных)	Дополнительная возможность (длина данных)
= (Равенство)	Не указывается: данные без знака. S: данные со знаком.	Не указывается: Данные из одного слова. L: данные двойной длины.
<> (Неравенство)		
< (Менее чем)		
<= (Менее чем или равно)		
> (Более чем)		
>= (Более чем или равно)		

Команды сравнения ввода без знака (т.е. команды, в которых отсутствует S) могут производить сравнение двоичных или двоично-десятичных данных без знака. Команды сравнения ввода со знаком (т.е. команды, в которых присутствует S) осуществляют сравнение двоичных или двоично-десятичных данных со знаком.

### Краткий обзор команд сравнения

В следующей ниже таблице приводятся функциональные коды, мнемонические изображения, наименования и функции всех 72-х команд сравнения. (Для сравнения данных из одного слова  $C1=S1$  и  $C2=S2$ ; для сравнения данных двойной длины  $C1=S1+1$ ,  $S1$  и  $C2=S2+1$ ,  $S2$ .)

Код	Мнемоническое изображение	Наименование	Функция
300	LD=	LOAD EQUAL	Истина, когда $C1=C2$
	AND=	AND EQUAL	
	OR=	OR EQUAL	
301	LD=L	LOAD DOUBLE EQUAL	
	AND=L	AND DOUBLE EQUAL	
	OR=L	OR DOUBLE EQUAL	
302	LA=S	LOAD SIGNED EQUAL	
	AND=S	AND SIGNED EQUAL	
	OR=S	OR SIGNED EQUAL	
303	LD=SL	LOAD DOUBLE SIGNED EQUAL	
	AND=SL	AND DOUBLE SIGNED EQUAL	
	OR=SL	OR DOUBLE SIGNED EQUAL	
305	LD<>	LOAD NOT EQUAL	Истина, когда $C1 \neq C2$
	AND<>	AND NOT EQUAL	
	OR<>	OR NOT EQUAL	



Код	Мнемоническое изображение	Наименование	Функция			
306	LD<>L	LOAD DOUBLE NOT EQUAL	Истина, когда C1<C2			
	AND<>L	AND DOUBLE NOT EQUAL				
	OR<>L	OR DOUBLE NOT EQUAL				
307	LD<>S	LOAD SIGNED NOT EQUAL				
	AND<>S	AND SIGNED NOT EQUAL				
	OR<>S	OR SIGNED NOT EQUAL				
308	LD<>SL	LOAD DOUBLE SIGNED NOT EQUAL				
	AND<>SL	AND DOUBLE SIGNED NOT EQUAL				
	OR<>SL	OR DOUBLE SIGNED NOT EQUAL				
310	LD<	LOAD LESS THAN		Истина, когда C1<C2		
	AND<	AND LESS THAN				
	OR<	OR LESS THAN				
311	LD<L	LOAD DOUBLE LESS THAN				
	AND<L	AND DOUBLE LESS THAN				
	OR<L	OR DOUBLE LESS THAN				
312	LD<S	LOAD SIGNED LESS THAN				
	AND<S	AND SIGNED LESS THAN				
	OR<S	OR SIGNED LESS THAN				
313	LD<SL	LOAD DOUBLE SIGNED LESS THAN				
	AND<SL	AND DOUBLE SIGNED LESS THAN				
	OR<SL	OR DOUBLE SIGNED LESS THAN				
315	LD<=	LOAD LESS THAN OR EQUAL			Истина, когда C1<=C2	
	AND<=	AND LESS THAN OR EQUAL				
	OR<=	OR LESS THAN OR EQUAL				
316	LD<=L	LOAD DOUBLE LESS THAN OR EQUAL				
	AND<=L	AND DOUBLE LESS THAN OR EQUAL				
	OR<=L	OR DOUBLE LESS THAN OR EQUAL				
317	LD<=S	LOAD SIGNED LESS THAN OR EQUAL				
	AND<=S	AND SIGNED LESS THAN OR EQUAL				
	OR<=S	OR SIGNED LESS THAN OR EQUAL				
318	LD<=SL	LOAD DOUBLE SIGNED LESS THAN OR EQUAL	Истина, когда C1<=C2			
	AND<=SL	AND DOUBLE SIGNED LESS THAN OR EQUAL				
	OR<=SL	OR DOUBLE SIGNED LESS THAN OR EQUAL				
320	LD>	LOAD GREATER THAN				Истина, когда C1>C2
	AND>	AND GREATER THAN				
	OR>	OR GREATER THAN				
321	LD>L	LOAD DOUBLE GREATER THAN				
	AND>L	AND DOUBLE GREATER THAN				
	OR>L	OR DOUBLE GREATER THAN				
322	LD>S	LOAD SIGNED GREATER THAN				
	AND>S	AND SIGNED GREATER THAN				
	OR>S	OR SIGNED GREATER THAN				
323	LD>SL	LOAD DOUBLE SIGNED GREATER THAN				
	AND>SL	AND DOUBLE SIGNED GREATER THAN				
	OR>SL	OR DOUBLE SIGNED GREATER THAN				
325	LD>=	LOAD GREATER THAN OR EQUAL		Истина, когда C1>=C2		
	AND>=	AND GREATER THAN OR EQUAL				
	OR>=	OR GREATER THAN OR EQUAL				

Код	Мнемоническое изображение	Наименование	Функция
326	LD>=L	LOAD DOUBLE GREATER THAN OR EQUAL	
	AND>=L	AND DOUBLE GREATER THAN OR EQUAL	
	OR>=L	OR DOUBLE GREATER THAN OR EQUAL	
327	LD>=S	LOAD SIGNED GREATER THAN OR EQUAL	
	AND>=S	AND SIGNED GREATER THAN OR EQUAL	
	OR>=S	OR SIGNED GREATER THAN OR EQUAL	
328	LD>=SL	LOAD DOUBLE SIGNED GREATER THAN OR EQUAL	
	AND>=SL	AND DOUBLE SIGNED GREATER THAN OR EQUAL	
	OR>=SL	OR DOUBLE SIGNED GREATER THAN OR EQUAL	

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг «Более чем»	>	ON, когда S1 > S2 и данные выражены одним словом. ON, когда S1+1, S1 > S2+1, S2, и данные выражены двойными словами. OFF в других случаях.
Флаг «Более чем или равно»	>=	ON, когда S1 S2 и данные выражены одним словом. ON, когда S1+1, S1 S2+1, S2, и данные выражены двойными словами. OFF в других случаях.
Флаг равенства	=	ON, когда S1 = S2 и данные выражены одним словом. ON, когда S1+1, S1 = S2+1, S2, и данные выражены двойными словами. OFF в других случаях.
Флаг неравенства	<>	ON, когда S1 S2 и данные выражены одним словом. ON, когда S1+1, S1 S2+1, S2, и данные выражены двойными словами. OFF в других случаях.
Флаг «Менее чем»	<	ON, когда S1 < S2 и данные выражены одним словом. ON, когда S1+1, S1 < S2+1, S2, и данные выражены двойными словами. OFF в других случаях.
Флаг «Менее чем или равно»	<=	ON, когда S1 S2 и данные выражены одним словом. ON, когда S1+1, S1 S2+1, S2, и данные выражены двойными словами. OFF в других случаях.
Флаг отрицательного значения	N	OFF

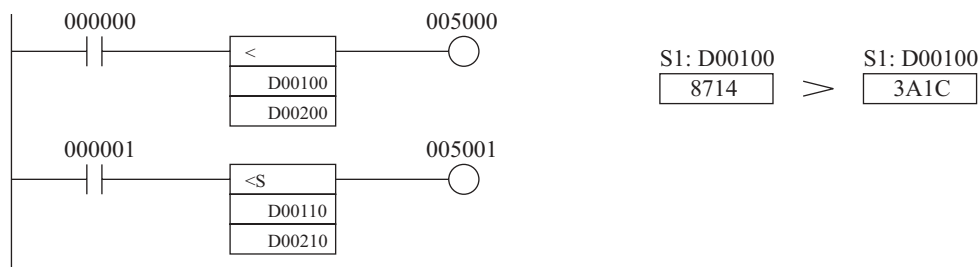
### Меры предосторожности

Команды сравнения ввода не могут использоваться в качестве команд, расположенных справа, т.е. между командой сравнения и правой линией шины должна находиться другая команда.

### Примеры

#### AND LESS THAN: AND<(310)

В следующем ниже примере, когда CIO 000001 находится в состоянии ON, содержания слов D00110 и D00210 сравниваются в виде двоичных данных со знаком. Если содержание D00110 меньше, чем содержание D00210, CIO 005001 переводится в состояние ON, и выполнение программы продолжается далее к следующей линии. Если содержание D00110 не меньше содержания D00210, остаток данной линии команд опускается, и выполнение программы переходит к следующей линии команд.

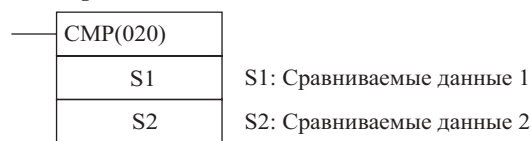


### 3-7-2 Команда сравнения COMPARE: CMP(020)

#### Назначение

По команде CMP(020) осуществляется сравнение двух величин в двоичном коде без знака (констант и или содержания указанных слов) и вывод результата сравнения в арифметические флаги во вспомогательной области.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	CMP(020).
	Выполняется один раз для дифференцирования вверх	Не поддерживается.
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		!CMP(020)

#### Применяемые области программы

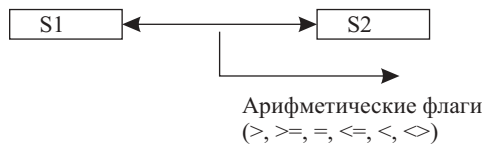
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Спецификации операндов

Область	S1	S2
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A000...A959	
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#0000...#FFFF (двоичн.)	
Регистры данных	DR0...DR15	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15	

**Описание**

По команде CMP(020) осуществляется сравнение двоичных данных без знака, содержащихся в S1 и S2, и вывод результата сравнения в арифметические флаги во вспомогательной области памяти (Более чем, Более чем или равно, Равно, Менее чем, Менее чем или равно, Флаг неравенства).

**Флаги состояния**

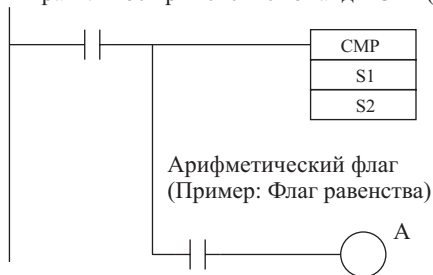
В следующей ниже таблице показано состояние арифметических флагов после выполнения команды CMP(020). (Знак «←» обозначает, что флаг может быть как в состоянии ON, так и в состоянии OFF.)

CMP(020)	Состояние флага					
	>	>=	=	<=	<	<>
S1 > S2	ON	ON	OFF	OFF	OFF	ON
S1 = S2	OFF	ON	ON	ON	OFF	OFF
S1, S2	OFF	OFF	OFF	ON	ON	ON

**Использование в программе результатов выполнения команды CMP(020)**

После выполнения команды CMP(020) результат выводится в арифметические флаги. Осуществляйте управление желаемым выводом или командой, находящейся справа, при помощи ответвления от условия выполнения, управляющего командой CMP(020), как показано на следующем рисунке. В данном случае, флаг равенства и вывод «А» переводятся в состояние ON, когда S1 = S2.

Правильное применение команды CMP(020)

**Использование в программе результатов выполнения команды CMP(020)**

Не допускается ввод другой команды между командой CMP(020) и командой, управляемой арифметическим флагом, так как вводимая таким образом команда может изменить состояние арифметического флага. В данном примере, результат выполнения команды В может изменить результат выполнения команды CMP(020).

Неправильное применение команды CMP(020)



Команда с указанием немедленной регенерации (!CMP(020)) может использоваться с указанными в S1 и/или S2 словами, распределяемыми внешним выводам. При выполнении команды !CMP(020), регенерация ввода выполняется для слова внешнего ввода, указанного в S1, и/или S2, и это обновленное значение используется для сравнения. (Немедленная регенерация не может выполняться для вводов, принадлежащих Высокоскоростным модулям ввода/вывода группы 2, или модулям, установленным в Slave- панели.)

**Флаги**

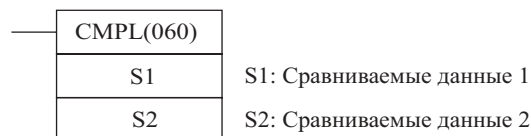
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг «Более чем»	>	ON, когда S1 > S2 и данные выражены одним словом. OFF в других случаях.
Флаг «Более чем или равно»	>=	ON, когда S1 ≥ S2 и данные выражены одним словом. OFF в других случаях.
Флаг равенства	=	ON, когда S1 = S2 и данные выражены одним словом. OFF в других случаях.
Флаг неравенства	<>	ON, когда S1 ≠ S2 и данные выражены одним словом. OFF в других случаях.
Флаг «Менее чем»	<	ON, когда S1 < S2 и данные выражены одним словом. OFF в других случаях.
Флаг «Менее чем или равно»	<=	ON, когда S1 ≤ S2 и данные выражены одним словом. OFF в других случаях.
Флаг отрицательного значения	N	OFF

**Меры предосторожности**

Не допускается ввод другой команды между командой CMP(020) и состоянием ввода, которое влияет на результат выполнения CMP(020), так как вводимая таким образом команда может изменить состояние арифметических флагов.

**3-7-3 Команда сравнения данных двойной длины DOUBLE COMPARE: CMPL(060)****Назначение**

Команда осуществляет сравнение двух двойных величин в двоичном коде без знака (константы и/ или содержание указанных слов) и вывод результата сравнения в арифметические флаги во вспомогательной области.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	CMPL(060).
	Выполняется один раз для дифференцирования вверх	Не поддерживается.
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

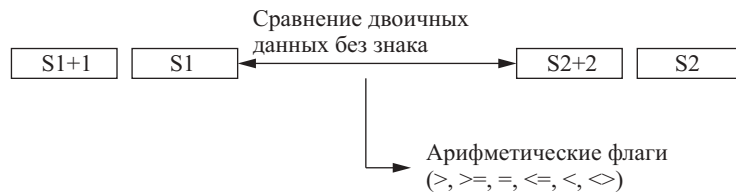
**Спецификации операндов**

Область	S1	S2
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	

Область	S1	S2
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFF (двоичн.)	
Регистры данных	DR0...DR15	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 , IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,(- -)IR0...,(- -)IR15	

### Описание

По команде CMPL(060) осуществляется сравнение двоичных данных без знаков, содержащихся в S1 +1, S1 и S2+1, S2, и вывод результата сравнения в арифметические флаги во вспомогательной области (Более чем, Более чем или равно, Равно, Менее чем, Менее чем или равно, Флаг неравенства).



### Состояния арифметических флагов

В следующей ниже таблице показано состояние арифметических флагов после выполнения команды CMPL(060). (Знак « $\leftrightarrow$ » обозначает, что флаг может быть как в состоянии ON, так и в состоянии OFF.)

CMP(020)	Состояние флага					
	>	>=	=	<=	<	<>
S1 + 1, S1 > S2 + 1, S2	ON	ON	OFF	OFF	OFF	ON
S1 + 1, S1 = S2 + 1, S2	OFF	ON	ON	ON	OFF	OFF
S1 + 1, S1 < S2 + 1, S2	OFF	OFF	OFF	ON	ON	ON

### Использование в программе результатов выполнения команды CMP(060)

После выполнения команды CMPL(060) результат выводится в арифметические флаги. Осуществляйте управление желаемым выводом или командой, находящейся справа, при помощи отвлечения от условия выполнения, управляющего командой CMPL(060), как показано на следующем рисунке. В данном случае, флаг равенства и вывод «A» переводятся в состояние ON, когда S1 + 1, S1 = S2 + 1, S2.

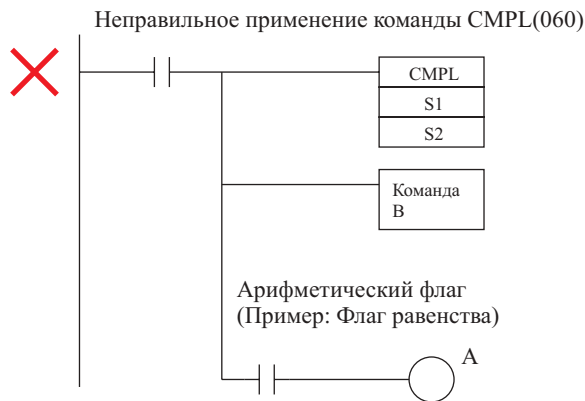
Правильное применение команды CMPL(060)



Правильное применение команды CMPL(060).

### Использование в программе результатов выполнения команды CMPL(060)

Не допускается ввод другой команды между командой CMPL(060) и командой, управляемой арифметическим флагом, так как вводимая таким образом команда может изменить состояние арифметического флага. В данном случае, результат выполнения команды B может изменить результат выполнения команды CMPL(060).



**Флаги**

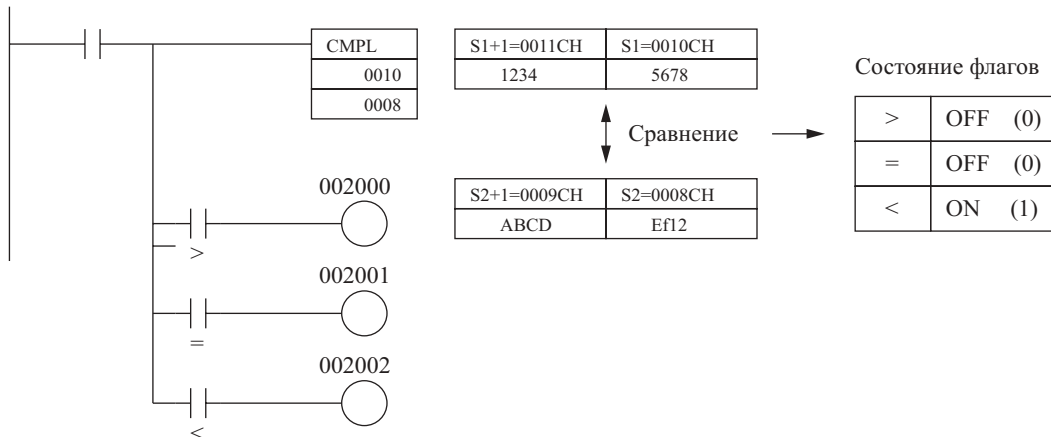
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг «Более чем»	>	ON, когда $S1 + 1, S1 > S2 + 1, S2$ . OFF в других случаях.
Флаг «Более чем или равно»	>=	ON, когда $S1 + 1, S1 \geq S2 + 1, S2$ . OFF в других случаях.
Флаг равенства	=	ON, когда $S1 + 1, S1 = S2 + 1, S2$ . OFF в других случаях.
Флаг неравенства	<>	ON, когда $S1 + 1, S1 \neq S2 + 1, S2$ . OFF в других случаях.
Флаг «Менее чем»	<	ON, когда $S1 + 1, S1 < S2 + 1, S2$ . OFF в других случаях.
Флаг «Менее чем или равно»	<=	ON, когда $S1 + 1, S1 \leq S2 + 1, S2$ . OFF в других случаях.
Флаг отрицательного значения	N	OFF

**Меры предосторожности**

Не допускается ввод другой команды между командой CMPL(060) и состоянием ввода, которое влияет на результат выполнения CMPL(060), так как вводимая таким образом команда может изменить состояние арифметических флагов.

**Пример**

В следующем примере, когда СЮ 000000 находится в состоянии ON, восьмизначные двоичные данные без знака, хранящиеся в СЮ 0011 и СЮ 0010, сравниваются с восьмизначными двоичными данными без знака, находящимися в СЮ 0009 и СЮ 0008. Результат сравнения выводится в арифметические флаги. Результат, записанный во флаги Более чем, Равенства, Менее чем, немедленно сохраняются в СЮ 000200 (Больше чем), СЮ 000201 (Равенство) и СЮ 000202 (Менее чем).



## 3-7-4 Команда сравнения двоичных данных со знаком SIGNED BINARY COMPARE: CPS(114)

**Назначение**

Команда осуществляет сравнение двух величин в двоичном коде без знака (константы и/ или содержание указанных слов) и вывод результата сравнения в арифметические флаги во вспомогательной области.

**Символ релейно-контактной схемы**

CPS(114)	
S1	S1: Сравнимые данные 1
S2	S2: Сравнимые данные 2

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	CPS(114).
	Выполняется один раз для дифференцирования вверх	Не поддерживается.
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	!CPS(114)	

**Применяемые области программы**

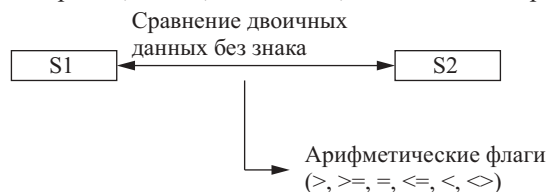
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	S1	S2
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A000...A959	
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#0000...#FFFF (двоичн.)	
Регистры данных	DR0...DR15	
Индексные регистры	-	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 , IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15(++ ,-(-)IR0...,(-)IR15	

**Описание**

По команде CPS(114) производится сравнение двоичных данных без знаков, содержащихся в S1 и S2, и вывод результата сравнения в арифметические флаги во вспомогательной области (Более чем, Более чем или равно, Равно, Менее чем, Менее чем или равно, Флаг неравенства).





**Примечание:** Команда CPS(114) обрабатывает данные, находящиеся в S1 и S2, как двоичные данные со знаком. Пределы изменения этих данных составляют: от 8000 до 7FFF (-32768 32767 в десятичном коде).

#### Состояния флагов

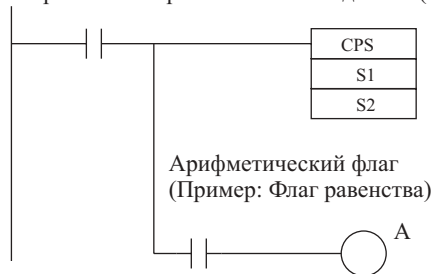
В следующей ниже таблице показано состояние арифметических флагов после выполнения команды CPS(114). (Знак «» обозначает, что флаг может быть как в состоянии ON, так и в состоянии OFF.)

CMP(020)	Состояние флага					
	>	>=	=	<=	<	<>
S1 > S2	ON	ON	OFF	OFF	OFF	ON
S1 = S2	OFF	ON	ON	ON	OFF	OFF
S1, S2	OFF	OFF	OFF	ON	ON	ON

#### Использование в программе результатов выполнения команды CPS(114)

После выполнения команды CPS(114) результат выводится в арифметические флаги. Осуществляйте управление желаемым выводом или командой, находящейся справа, при помощи ответвления от условия выполнения, управляющего командой CPS(114), как показано на следующем рисунке. В данном случае, флаг равенства и вывод «A» переводятся в состояние ON, когда S1 = S2.

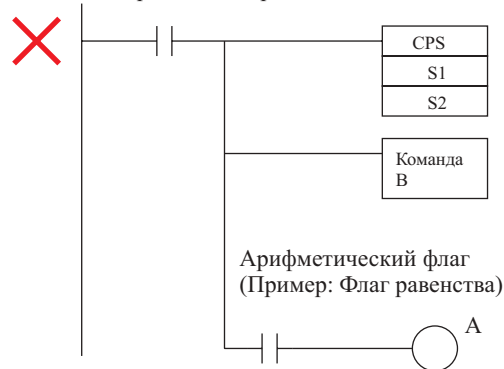
Правильное применение команды CPS(114)



#### Использование в программе результатов выполнения команды CPS(114)

Не допускается ввод другой команды между командой CPS(114) и командой, управляемой арифметическим флагом, так как вводимая таким образом команда может изменить состояние арифметического флага. В данном случае, результат выполнения команды B может изменить результат выполнения команды CPS(114).

Неправильное применение команды CPS(114)



Команда с указанием немедленной регенерации (!CMP(020)) может использоваться с указанными в S1 и/или S2 словами, распределяемыми внешним выводам. При выполнении команды !CMP(020), регенерация ввода выполняется для слова внешнего ввода, указанного в S1, и/или S2, и это обновленное значение используется для сравнения. (Немедленная регенерация не может выполняться для вводов, принадлежащих Модулям группы 2, или модулям, установленным в Slave-панели.)

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг «Более чем»	>	ON, когда S1 > S2. OFF в других случаях.

Наименование	Обозначение	Функционирование
Флаг «Более чем или равно»	>=	ON, когда S1 ≥ S2. OFF в других случаях.
Флаг равенства	=	ON, когда S1 = S2. OFF в других случаях.
Флаг неравенства	<>	ON, когда S1 ≠ S2. OFF в других случаях.
Флаг «Менее чем»	<	ON, когда S1 < S2. OFF в других случаях.
Флаг «Менее чем или равно»	<=	ON, когда S1 ≤ S2. OFF в других случаях.
Флаг отрицательного значения	N	OFF

### Меры предосторожности

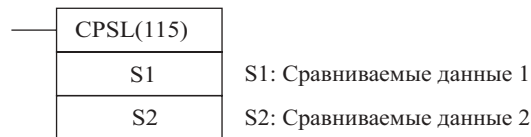
Не допускается ввод другой команды между командой CPS(114) и состоянием ввода, которое влияет на результат выполнения CPS(114), так как вводимая таким образом команда может изменить состояние арифметических флагов.

### 3-7-5 Команда сравнения данных двойной длины DOUBLE COMPARE: CPSL(115)

#### Назначение

Команда осуществляет сравнение двух двойных чисел в двоичном коде со знаком (константы и/или содержание указанных слов) и вывод результата сравнения в арифметические флаги во вспомогательной области.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	CPSL(115).
	Выполняется один раз для дифференцирования вверх	Не поддерживается.
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

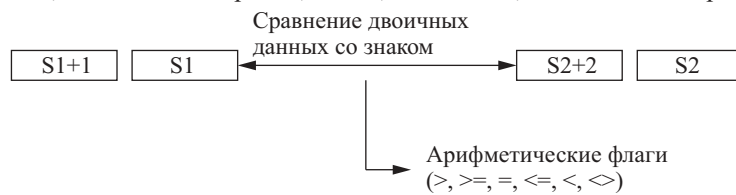
#### Спецификации операндов

Область	S1	S2
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_ 00000...En_ 32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_ 00000...@En_ 32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_ 00000...*En_ 32767 (n = 0...C)	

Область	S1	S2
Константы	#00000000...#FFFFFFF (двоичн.)	
Регистры данных	DR0...DR15	
Индексные регистры	-	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 , IR15 DR0...DR15, IR0... IR15 ,IR0(++)... IR15(++) ,(- -)IR0...,(- -)IR15	

**Описание**

По команде CPSL(115) осуществляется сравнение двоичных данных со знаком, содержащихся в S1 +1, S1 и S2+1, S2, и вывод результата сравнения в арифметические флаги во вспомогательной области (Более чем, Более чем или равно, Равно, Менее чем, Менее чем или равно, Флаг неравенства).



**Примечание:** Команда CPSL(115) обрабатывает данные, находящиеся в S1 и S2, как двоичные данные двойной длины со знаком. Пределы изменения этих данных составляют: от 80000000 до 7FFF FFFF (-2147483648...2147483647 в десятичном коде).

**Состояния арифметических флагов**

В следующей ниже таблице показано состояние арифметических флагов после выполнения команды CPSL(115). (Знак «←» обозначает, что флаг может быть как в состоянии ON, так и в состоянии OFF.)

CPSL(115)	Состояние флага					
	>	>=	=	<=	<	<>
S1 + 1, S1 > S2 + 1, S2	ON	ON	OFF	OFF	OFF	ON
S1 + 1, S1 = S2 + 1, S2	OFF	ON	ON	ON	OFF	OFF
S1 + 1, S1 < S2 + 1, S2	OFF	OFF	OFF	ON	ON	ON

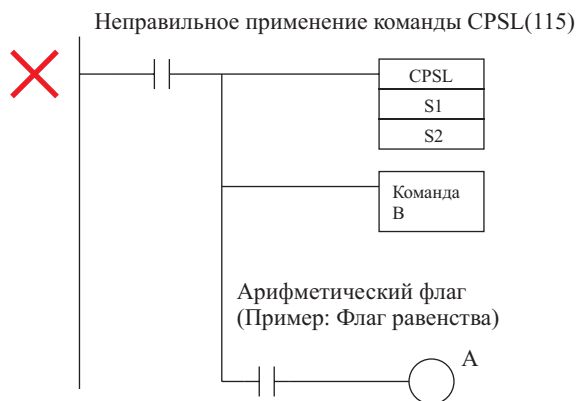
**Использование в программе результатов выполнения команды CPSL(115)**

После выполнения команды CPSL(115) результат выводится в арифметические флаги. Осуществляйте управление желаемым выводом или командой, находящейся справа, при помощи ответвления от условия выполнения, управляющего командой CPSL(115), как показано на следующем рисунке. В данном случае, флаг равенства и вывод «А» переводятся в состояние ON, когда S1 + 1, S1 = S2 + 1, S2.

Правильное применение команды CPSL(115)

**Использование в программе результатов выполнения команды CPSL(115)**

Не допускается ввод другой команды между командой CPSL(115) и командой, управляемой арифметическим флагом, так как вводимая таким образом команда может изменить состояние арифметического флага. В данном случае, результат выполнения команды В может изменить результат выполнения команды CPSL(115).



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг «Более чем»	>	ON, когда $S1 + 1, S1 > S2 + 1, S2$ . OFF в других случаях.
Флаг «Более чем или равно»	$\geq$	ON, когда $S1 + 1, S1 \geq S2 + 1, S2$ . OFF в других случаях.
Флаг равенства	=	ON, когда $S1 + 1, S1 = S2 + 1, S2$ . OFF в других случаях.
Флаг неравенства	$\neq$	ON, когда $S1 + 1, S1 \neq S2 + 1, S2$ . OFF в других случаях.
Флаг «Менее чем»	<	ON, когда $S1 + 1, S1 < S2 + 1, S2$ . OFF в других случаях.
Флаг «Менее чем или равно»	$\leq$	ON, когда $S1 + 1, S1 \leq S2 + 1, S2$ . OFF в других случаях.
Флаг отрицательного значения	N	OFF

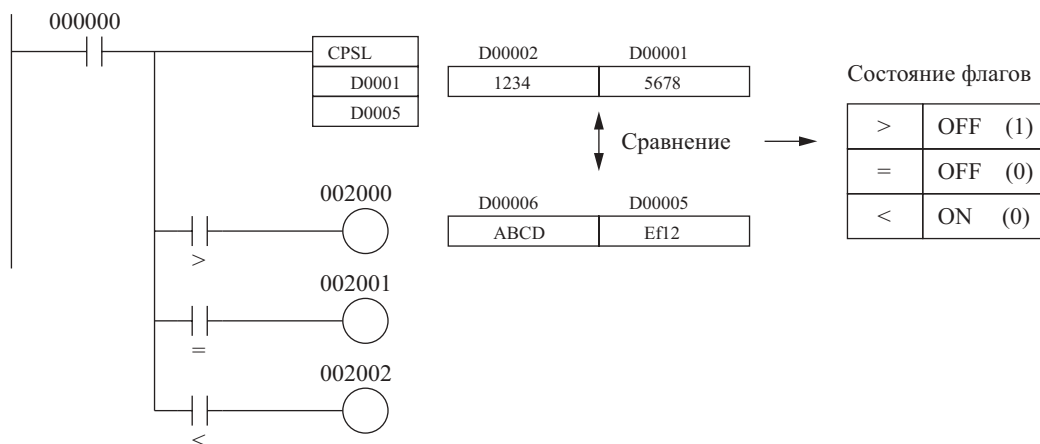
### Меры предосторожности

Не допускается ввод другой команды между командой CPSL(115) и состоянием ввода, которое влияет на результат выполнения CPSL(115), так как вводимая таким образом команда может изменить состояние арифметических флагов.

### Пример

В следующем примере, когда CIO 000000 находится в состоянии ON, восьмизначные двоичные данные без знака, хранящиеся в D00002 и D00001, сравниваются с восьмизначными двоичными данными без знака, находящимися в D00006 и D00005. Результат сравнения выводится в арифметические флаги.

- Если содержание D00002 и D00001 больше, чем содержание D00006 и D00005, флаг «Более чем» переводится в состояние ON, способствуя переводу CIO 002000 также в состояние ON.
- Если содержание D00002 и D00001 равно содержанию D00006 и D00005, флаг равенства переводится в состояние ON, способствуя переводу CIO 002001 также в состояние ON.
- Если содержание D00002 и D00001 меньше, чем содержание D00006 и D00005, флаг «Менее чем» переводится в состояние ON, способствуя переводу CIO 002002 также в состояние ON.

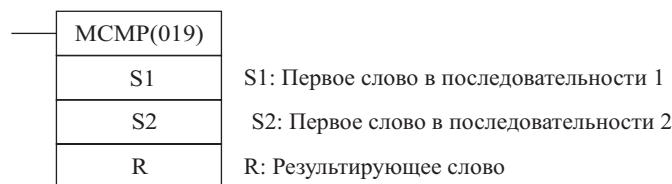


### 3-7-6 Команда сравнения последовательностей слов MULTIPLE COMPARE: MCMP(019)

#### Назначение

По команде MCMP(019) осуществляется сравнение шестнадцати последовательных слов с другими шестнадцатью последовательными словами и перевод в состояние ON соответствующего бита в результирующем слове, когда слова не равны.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	MCMP(019)
	Выполняется один раз для дифференцирования вверх	@MCMP(019)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

##### S1: Первое слово в последовательности 1

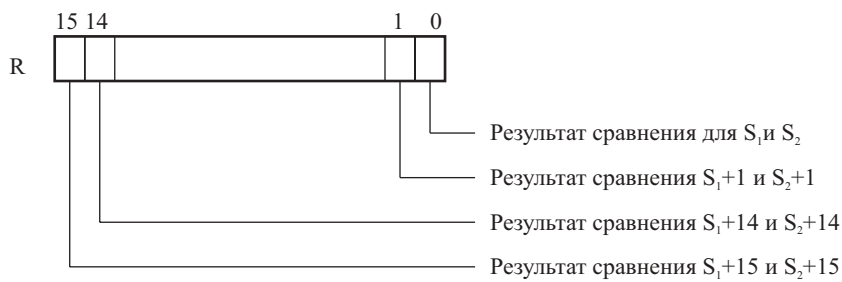
Указывает на первое слово из последовательности слов, состоящей из 16-ти слов. S1 и S1+15 должны находиться в одной области данных.

##### S2: Первое слово в последовательности 2

Указывает на первое слово во второй последовательности из 16-ти слов. S2 и S2+15 должны находиться в одной области данных.

##### R: Результирующее слово.

Каждый из битов слова R содержит результат сравнения двух слов из последовательностей, состоящих из 16-ти слов. Бит n слова R (n= 00 15) содержит результат сравнения слов S1+n и S2+n.



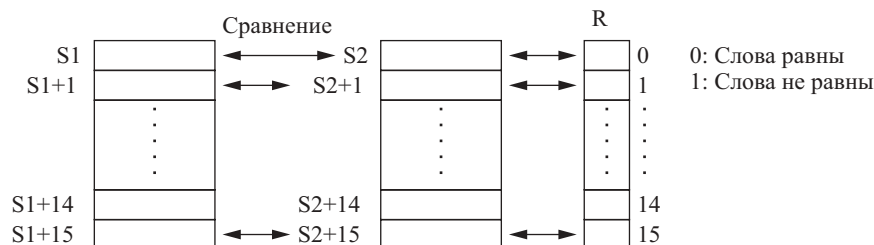
### Спецификации операндов

Область	S1	S2	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6128	CIO 0000...CIO 6143	
Рабочая область	W000...W496	W000...W511	
Область удержания бита (Holding bit area)	H000...H496	H000...H511	
Область вспомогательных битов	A000...A944	A448...A959	
Область таймера	T0000...T4080	T0000...T4095	
Область счетчика	C0000...C4080	C0000...C4095	
Область DM	D00000...D32752	D00000...D32767	
Область EM, не содержащая банков	E00000...E32752	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32752 (n = 0...C)	En_00000...En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	-		
Регистры данных	-		
Регистры данных		DR0...DR15	
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047,IR0 -2048...+2047, IR15 DR0...DR15, IR0...IR15 ,IR0+(++)..., IR15+(++) ,-(-)IR0...,(-(-)IR15		

### Описание

По команде MPCM(019) осуществляется сравнение содержания 16-ти слов от S1 до S1+15 с содержанием 16-ти слов от S2 до S2+15. Соответствующий бит в слове R переводится в состояние ON, если последовательности слов отличаются друг от друга.

Содержание слова S1 сравнивается с содержанием слова S2, содержание слова S1+1 сравнивается с содержанием слова S2+1... содержание слова S1+15 сравнивается с содержанием слова S2+15. Бит n слова R переводится в состояние OFF, когда содержание слова S1+n равно содержанию слова S2+n. Бит n слова R переводится в состояние ON, когда содержания слов не равны. Когда содержание всех 16-ти пар слов одинаково, после выполнения команды флаг равенства переводится в состояние ON.



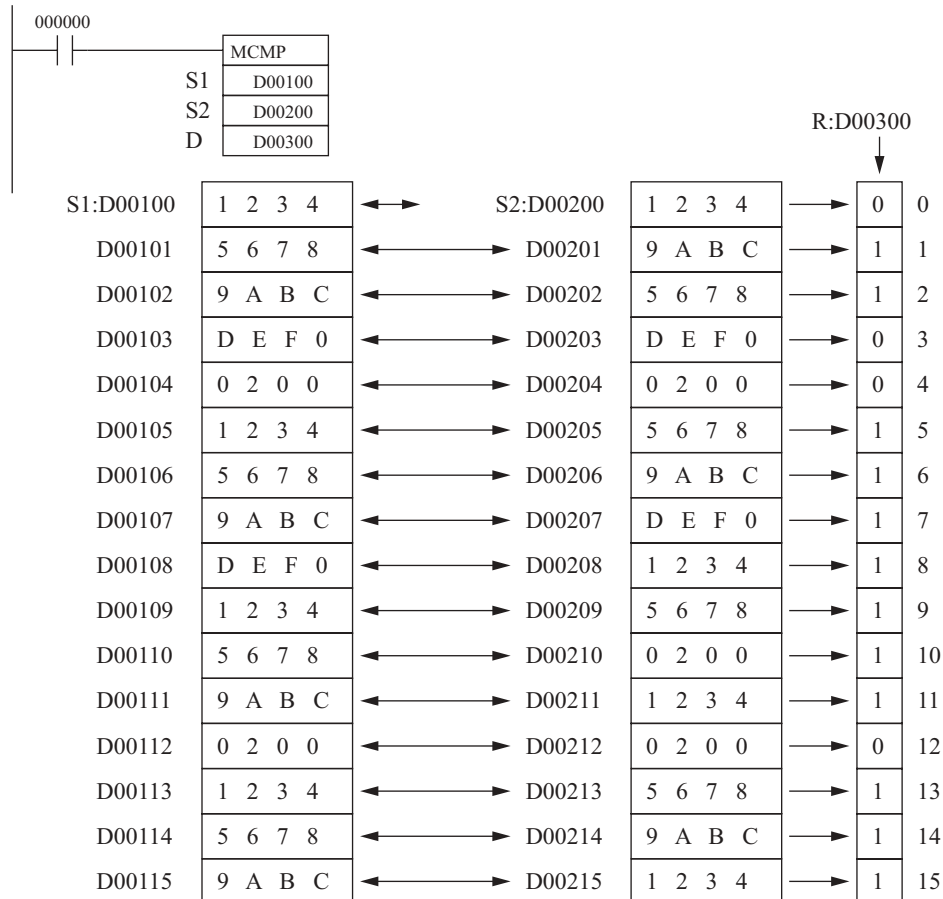
### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF

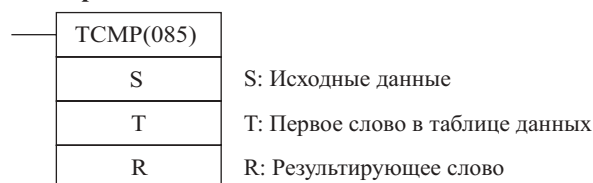
Наименование	Обозначение	Функционирование
Флаг равенства	=	ON, когда результирующим словом является 0000. (Две последовательности слов по 16 слов содержат одинаковые данные.) OFF во всех прочих случаях.

**Пример**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, по команде MCMP(019) производится сравнение слов от D00100 до D00115 со словами от D00200 до D00215. Соответствующий бит в D00300 переводится в состояние ON, когда слова не равны друг другу.

**3-7-7 Команда сравнения табличных данных TABLE COMPARE: TCMP(085)****Назначение**

По команде TCMP(085) осуществляется сравнение исходных данных с содержанием шестнадцати последовательных слов и перевод в состояние ON соответствующего бита в результирующем слове, когда содержание слов одинаково.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	TCMP(085)
	Выполняется один раз для дифференцирования вверх.	@ TCMP(085)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

Области блоков программы	Области шагов программы	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

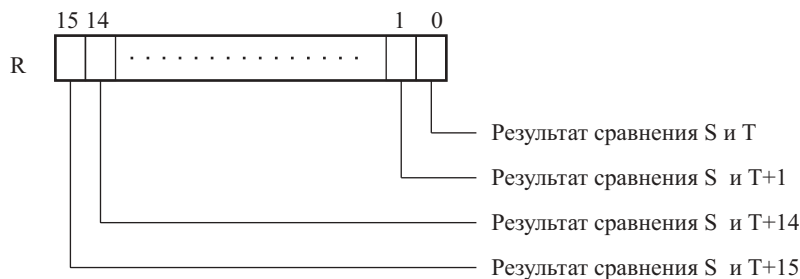
**Операнды****T: Первое слово в таблице**

Указывает на первое слово таблицы, состоящей из 16-ти слов. T и T+15 должны находиться в одной области данных.

**R: Результирующее слово.**

Каждый из битов слова R содержит результат равенства исходных данных S со словами таблицы, состоящей из 16-ти слов. Бит n слова R (n = 00 15) содержит результат сравнения слова S и T+n.

T	Данные сравнения 0
T+1	Данные сравнения 1
T+15	Данные сравнения 15

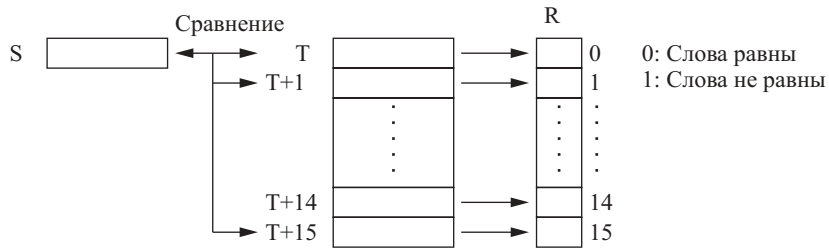
**Спецификации операндов**

Область	S	T	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6128	CIO 0000...CIO 6143
Рабочая область	W000...W511	W000...W496	W000...W511
Область удержания бита (Holding bit area)	H000...H511	H000...H496	H000...H511
Область вспомогательных битов	A448...A959	A000...A944	A448...A959
Область таймера	T0000...T4095	T0000...T4080	T0000...T4095
Область счетчика	C0000...C4095	C0000...C4080	C0000...C4095
Область DM	D00000...D32767	D00000...D32752	D00000...D32767
Область EM, не содержащая банков	E00000...E32767	E00000...E32752	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32752 (n=0...C)	En_00000...En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF	---	
Регистры данных	DR0...DR15	---	DR0...DR15
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047, IR0...-2048...+2047, IR15 DR0...DR15, IR0...IR15 ,IR0+(++)... , IR15+(++) ,-(-)IR0...,-(-)IR15		

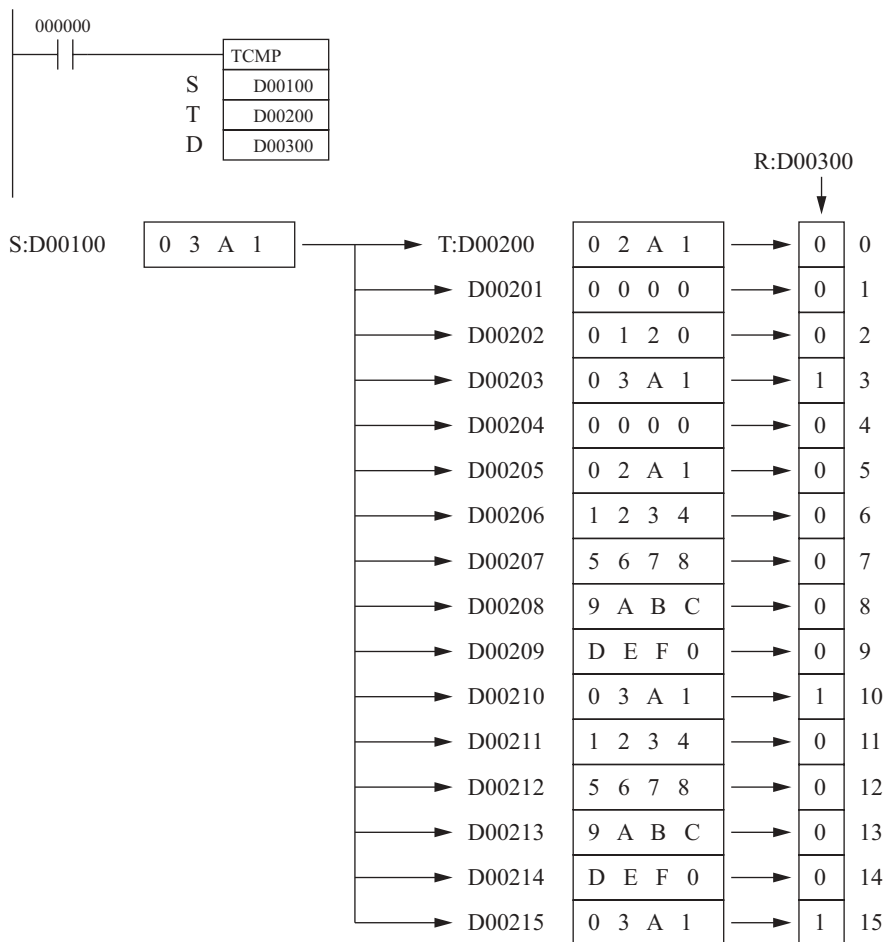
**Описание**

По команде TCMR(085) производится сравнение исходных данных (S) с содержанием 16-ти слов таблицы от T до T+15. Соответствующий бит n слова R переводится в состояние ON, когда содержание T+n равно S, и переводится в состояние OFF, когда слова отличаются друг от друга.





Наименование	Обозначение	Функционирование



## 3-7-8 Команда сравнения блока BLOCK COMPARE: BCMP(068)

**Назначение**

По команде BCMP(068) осуществляется сравнение исходных данных с содержанием шестнадцати диапазонов слов (определяемых соответственно 16-ю нижними и 16-ю верхними пределами) и перевод в состояние ON соответствующего бита в результирующем слове, когда исходные данные находятся в заданном пределе.

**Символ релейно-контактной схемы**

BCMP(068)	
S	S: Исходные данные
B	T: Первое слово в блоке
R	R: Результирующее слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	BCMP(068)
	Выполняется один раз для дифференцирования вверх	@ BCMP(068)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

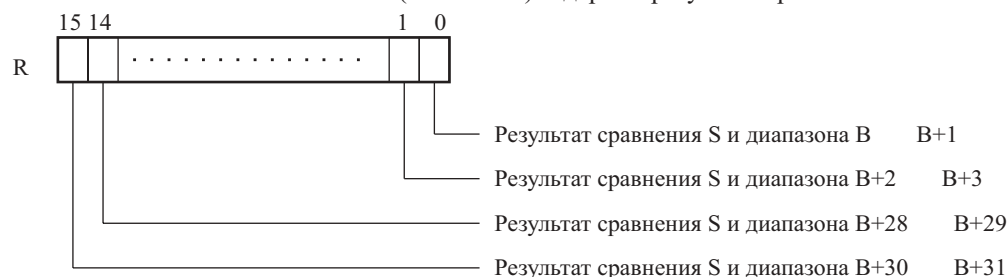
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****В: Первое слово блока**

Указывает на первое слово блока, состоящего из 32-х слов (16 пар слов нижнего/верхнего предела). В и В+15 должны находиться в одной области данных.

**R: Результирующее слово.**

Каждый из битов слова R содержит результат равенства S с одним из 16-ти пределов, определяемых блоком из 32-х слов. Бит n слова R ( $n = 00 \dots 15$ ) содержит результат сравнения слова S и пары слов № n.

**Спецификации операндов**

Область	S	B	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6112	CIO 0000...CIO 6143
Рабочая область	W000...W511	W000...W480	W000...W511
Область удержания бита (Holding bit area)	H000...H511	H000...H480	H000...H511
Область вспомогательных битов	A448...A959	A000...A928	A448...A959
Область таймера	T0000...T4095	T0000...T4064	T0000...T4095
Область счетчика	C0000...C4095	C0000...C4064	C0000...C4095
Область DM	D00000...D32767	D00000...D32736	D00000...D32767
1 Область EM, не содержащая банков	E00000...E32767	E00000...E32736	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32736 (n = 0...C)	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		

Область	S	B	R
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF	–	
Регистры данных	DR0...DR15	–	DR0...DR15
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0...-2048...+2047 , IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-)IR15		

**Описание**

По команде ВСМР(068) осуществляется сравнение исходных данных (S) с содержанием 16-ти диапазонов, определяемых нижними и верхними предельными значениями от B до B+31. Первое слово в каждой паре (B+2n) указывает нижний предел, а второе (B+2n+1) – верхний предел диапазона n (n=0...15). Когда S находится в одном из указанных диапазонов (обозначенных верхними и нижними пределами), соответствующий бит слова R переводится в состояние ON. Все прочие биты слова R переводятся в состояние OFF.

$B \leq B+1$	бит 00 слова R
$B+2 \leq B+3$	бит 01 слова R
$B+4 \leq B+5$	бит 02 слова R
$B+6 \leq B+7$	бит 03 слова R
$B+8 \leq B+9$	бит 04 слова R
$B+10 \leq B+11$	бит 05 слова R
$B+12 \leq B+13$	бит 06 слова R
$B+14 \leq B+15$	бит 07 слова R
$B+16 \leq B+17$	бит 08 слова R
$B+18 \leq B+19$	бит 09 слова R
$B+20 \leq B+21$	бит 10 слова R
$B+22 \leq B+23$	бит 11 слова R
$B+24 \leq B+25$	бит 12 слова R
$B+26 \leq B+27$	бит 13 слова R
$B+28 \leq B+29$	бит 14 слова R
$B+30 \leq B+31$	бит 15 слова R

Например, бит 00 слова R переводится в состояние ON, когда S находится в первом диапазоне ( $B \leq S \leq B+1$ ). Бит 01 слова R переводится в состояние ON, когда S находится во втором диапазоне ( $B+2 \leq S \leq B+3$ ),..., Бит 15 слова R переводится в состояние ON, когда S находится в пятнадцатом диапазоне ( $B+30 \leq S \leq B+31$ ). Все прочие биты слова R переводятся в состояние OFF.

**Флаги**

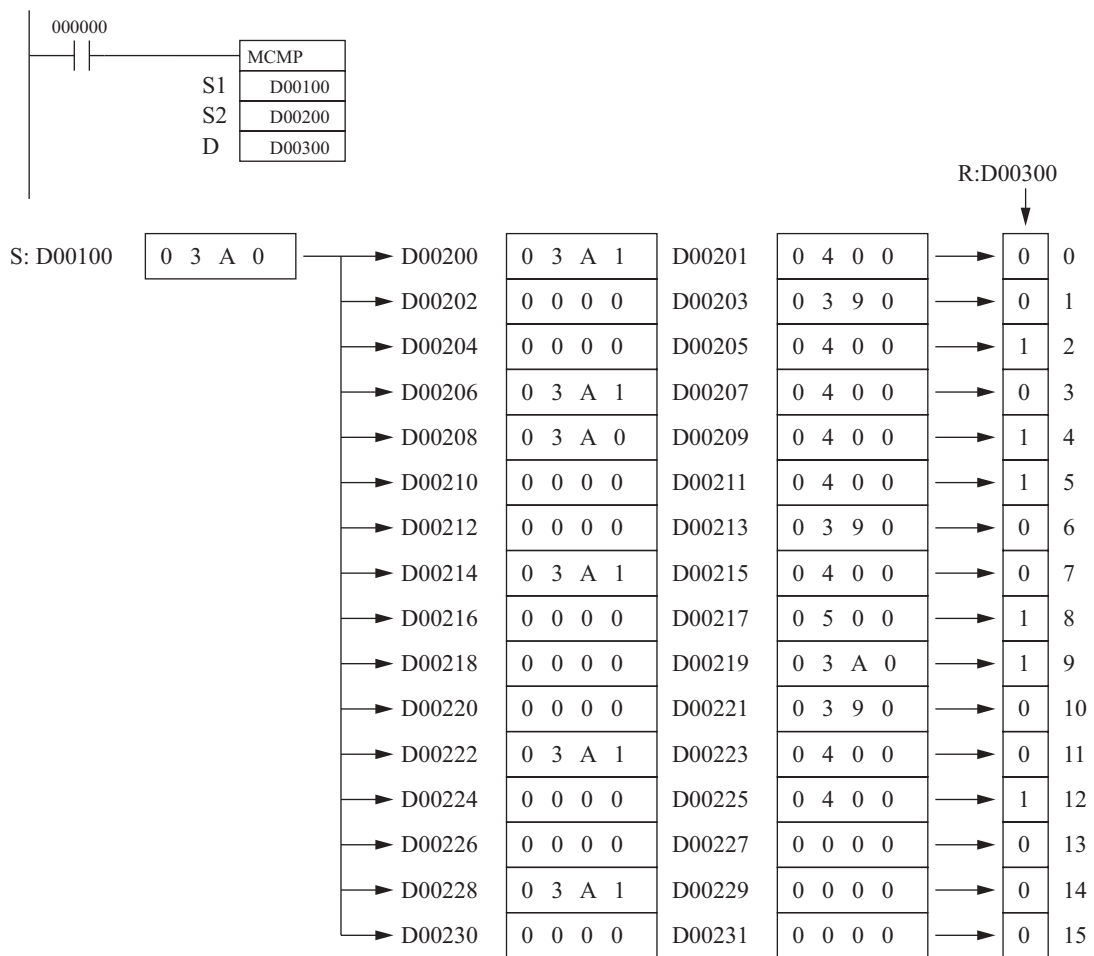
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	ON, когда результирующим словом является 0000. (S не находится ни в одном из диапазонов.) OFF во всех прочих случаях.

**Меры предосторожности**

В случае, когда нижний предел больше верхнего предела, ошибка не определяется, однако в соответствующий бит слова R заносится 0 (слово находится за пределами диапазона.)

**Пример**

В следующем ниже примере, когда СЮ 000000 находится в состоянии ON, по команде ВСМР(068) осуществляется сравнение слова D00100 с содержанием слов в 16-ти диапазонах, определяемых словами от D00200 до D00231. Соответствующий бит в D00300 переводится в состояние ON, когда S находится в указанном диапазоне, и переводится в состояние OFF, когда S находится за пределами диапазона.

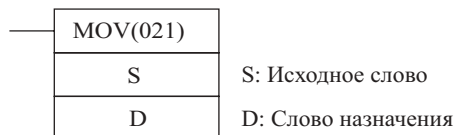


## 3-8 Команды перемещения данных

## 3-8-1 Команда MOVE: MOV(021)

**Назначение**

По команде MOV(021) осуществляется перемещение одного слова данных в указанное слово.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	MOV(021)
	Выполняется один раз для дифференцирования вверх	@ MOV(021)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается
Комбинированная модификация	Немедленная регенерация выполняется один раз для дифференцирования вверх.	! @MOV(021)

**Применяемые области программы**

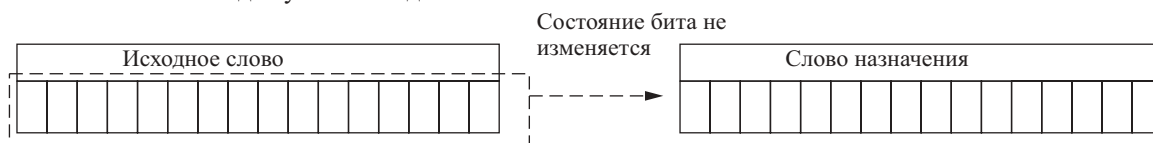
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A448...A959	A448...A959
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#0000...#FFFF (двоичные)	—
Регистры данных	DR0...DR15	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 , IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-(-)IR15	

**Описание**

По команде MOV(021) осуществляется перемещение S в D. Если S является константой, это значение может использоваться для установки данных.



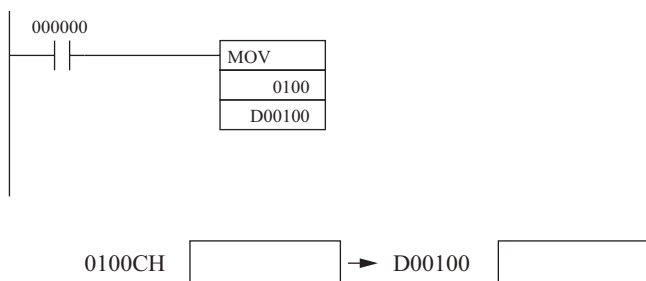
Команда MOV(021) обладает модификацией с немедленной регенерацией (!MOV(021)). Биты внешнего ввода могут указываться для S, и для D могут указываться биты внешнего вывода. Биты ввода, используемые для S, подвергаются регенерации непосредственно перед выполнением команды, а биты вывода используемые для D, - непосредственно после выполнения команды, если эти биты не распределяются Модулям группы 2, Специальным модулям ввода/вывода или Модулям, установленным в Slave-панелях SYSMAC BUS.

**Флаги**

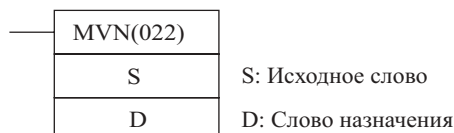
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	ON, когда перемещаемые данные равны 0000. OFF в других случаях.
Флаг отрицательного значения	N	ON, когда старший бит перемещаемых данных равен 1. OFF в других случаях.

**Пример**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, содержание CIO 0100 записывается в D00100.

**3-8-2 Команда MOVE NOT: MVN(022)****Назначение**

По команде MVN(022) осуществляется перемещение дополнения к слову данных в указанное слово.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	MVN(022)
	Выполняется один раз для дифференцирования вверх	@ MVN(022)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

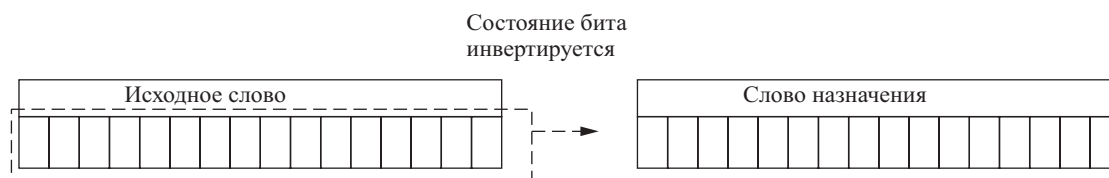
**Спецификации операндов**

Область	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A448...A959	A448...A959
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	

Область	S	D
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#0000...#FFFF (двоичные)	—
Регистры данных	DR0...DR15	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 , IR15 DR0...DR15, IR0...IR15 ,IR0+(++)... ,IR15+(++) ,-(-)IR0...,-(-)IR15	

**Описание**

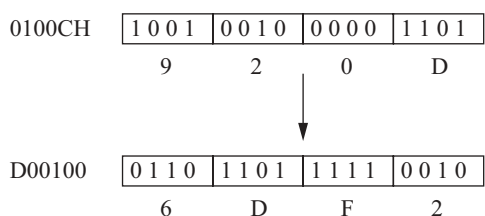
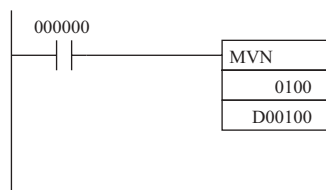
По команде MVN(022) осуществляется инвертирование битов в слове S и запись результата в D. Содержание S остается неизменным.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержанием D является 0000. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит D равен 1. OFF в других случаях.

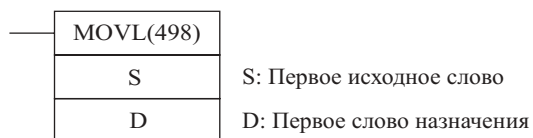
**Пример**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, содержание CIO 0100 инвертируется и результат записывается в D00100.

**3-8-3 Команда перемещения двойного слова DOUBLE MOVE: MOVL(489)****Назначение**

По команде MVL(489) осуществляется перемещение двух слов данных в указанные слова.

## Символ релейно-контактной схемы



## Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	MVL(489)
	Выполняется один раз для дифференцирования вверх.	@ MVL(489)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается.	

## Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

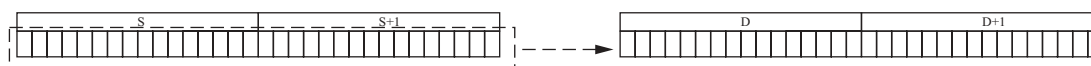
## Спецификации операндов

Область	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A448...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFF (двоичные)	—
Регистры данных	—	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047,IR0 -2048...+2047, IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15	

## Описание

По команде MOVL(498) осуществляется перемещение данных S+1 и S в D+1 и D. Если S+1 и S являются константами, эти значения могут использоваться для установки данных.

Состояние бита не  
изменяется



## Флаги

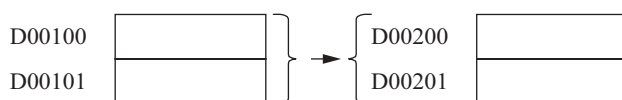
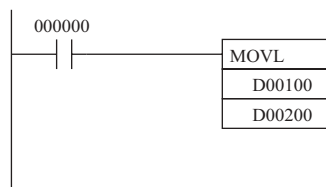
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF



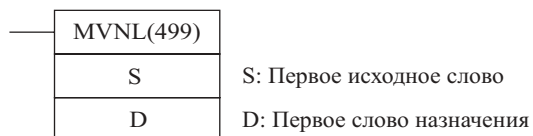
Наименование	Обозначение	Функционирование
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержанием D+1 и D является 0000 0000. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит D+1 равен 1. OFF в других случаях.

**Пример**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, содержание D00101 и D00100 записывается в D00201 и D00200.

**3-8-4 Команда передачи двойного дополнения DOUBLE MOVE NOT: MVNL(499)****Назначение**

По команде MVNL(499) осуществляется передача дополнения к двум словам данных в указанные слова.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	MVNL(499)
	Выполняется один раз для дифференцирования вверх	@ MVNL(499)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

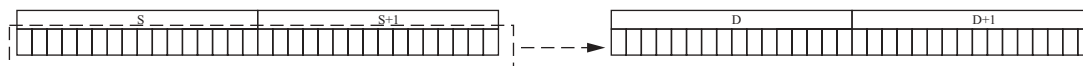
Область	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)	

Область	S	D
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFF (двоичные)	—
Регистры данных	—	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 , IR15 DR0...DR15, IR0...IR15 ,IR0+(++)... ,IR15+(++) ,-(-)IR0...,-(-)IR15	

**Описание**

По команде MVNL(499) осуществляется инвертирование S+1 и S и перемещение результата в D+1 и D. Содержание S+1 и S при этом не изменяется.

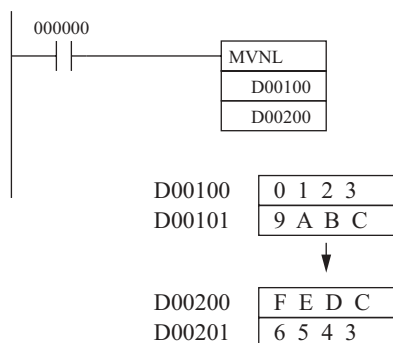
Состояние бита  
изменяется

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержанием D+1 и D является 0000 0000. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит D+1 равен 1. OFF в других случаях.

**Пример**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, содержание D00101 и D00100 инвертируется и результат записывается в D00201 и D00200. (Первоначальное содержание в D00101 и D00100 не изменяется.)

**3-8-5 Команда перемещения бита MOVE BIT: MOVNB(082)****Назначение**

По команде MOVNB(082) осуществляется перемещение указанного бита.

**Символ релейно-контактной схемы**

MOVNB(082)	
S	S: Исходное слово или данные
C	C: Контрольное слово
D	D: Слово назначения

**Модификации**

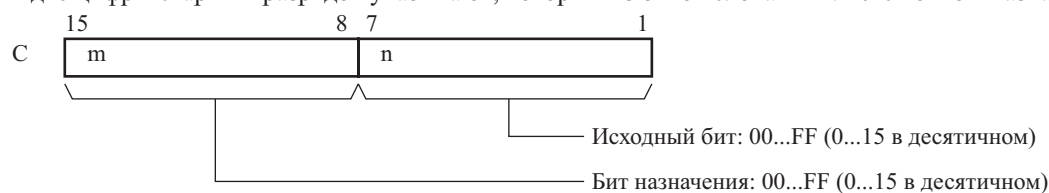
Модификации	Выполняется в каждом цикле для состояния ON.	MOVВ(082)
	Выполняется один раз для дифференцирования вверх	@ MOVВ(082)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****С: управляющее слово**

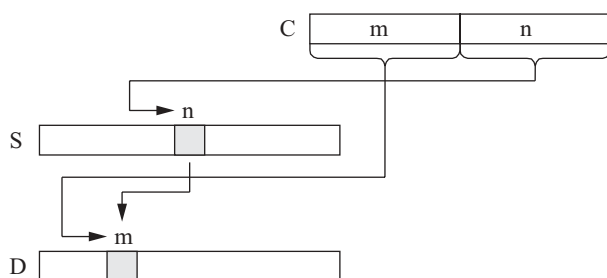
Две цифры младших разрядов слова С указывают, который из битов слова S является исходным битом, и две цифры старших разрядов указывают, который из битов слова D является битом назначения.

**Спецификации операндов**

Область	S	C	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000.E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF (двоичные)	Только указанные значения	—
Регистры данных	DR0...DR15		
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++),...,IR15(++) ,(- -)IR0...,-(- -)IR15		

**Описание**

По команде MOVВ(082) осуществляется перемещение указанного бита (n) из слова S в указанный бит (m) в слове D. Другие биты в слове назначения D не изменяются.



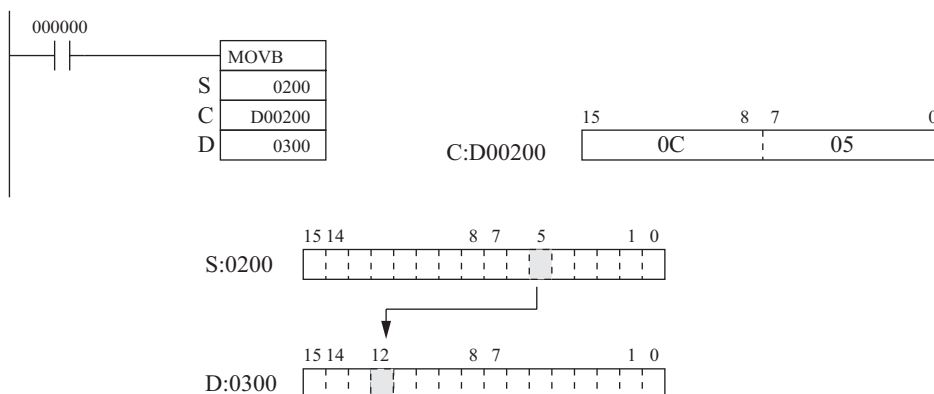
**Примечание:** Для перезаписи определенного бита внутри одного слова, в качестве *S* и *D* должно указываться одно слово.

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда две цифры старшего разряда и две цифры младшего разряда в слове <i>C</i> не входят в указанный диапазон от 0 до 0F. OFF в других случаях.

#### Пример

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, пятый бит исходного слова (CIO 0200) записывается в двенадцатый бит слова назначения (CIO 0300) в соответствии со значением управляющего слова 0C05.

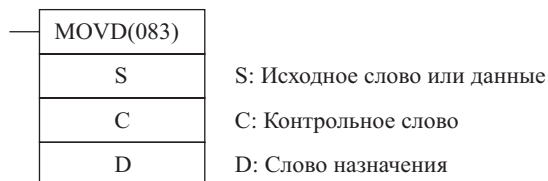


### 3-8-6 Команда перемещения числа MOVE DIGIT: MOVD(083)

#### Назначение

По команде MOVD(083) осуществляется перемещение указанной цифры или нескольких цифр. (Каждая цифра состоит из четырех битов.)

#### Символ релейно-контактной схемы



#### Модификации

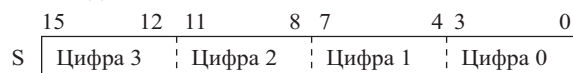
Модификации	Выполняется в каждом цикле для состояния ON.	MOVD(083)
	Выполняется один раз для дифференцирования вверх	@ MOVD(083)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

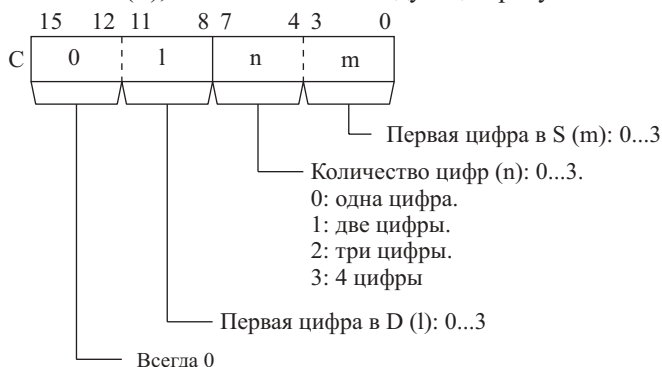
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****S: Исходное слово**

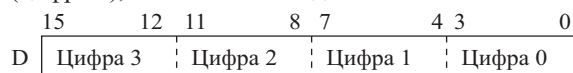
Исходные цифры читаются справа налево, возвращаясь к цифре (цифра 0) младшего разряда, если это необходимо.

**C: управляющее слово**

Три первые цифры слова C указывают на первую исходную цифру (m), вторые три цифры показывают количество цифр, подлежащих передаче (n), затем идут три цифры, показывающие первую цифру назначения (L), как показано на следующем рисунке.

**D: слово назначения**

Цифры в слове назначения записываются справа налево, возвращаясь к цифре младшего разряда (цифра 0), если это необходимо.

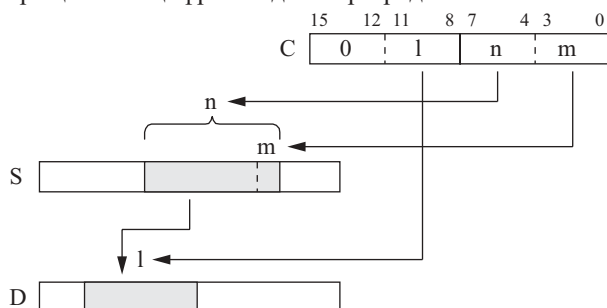
**Спецификации операндов**

Область	S	C	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF (двоичные)	Только указанные значения	—
Регистры данных	DR0...DR15		
Индексные регистры	—		

Область	S	C	D
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047,IR0 -2048...+2047, IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15		

**Описание**

По команде MOVD(083) осуществляется запись n цифр слова S (начиная с цифры m) в D (начиная с цифры l). Изменяются только указанные цифры. Другие цифры в слове назначения D не изменяются. Если количество читаемых или записываемых цифр превышает размер S или D, команда MOVD(083) возвращается к цифре младшего разряда в этом же слове.



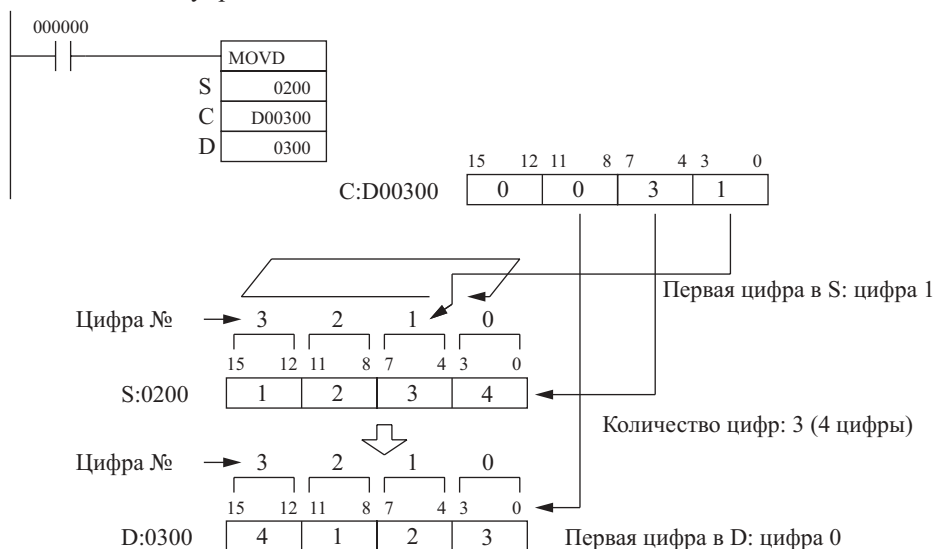
**Примечание:** Для перезаписи определенного бита внутри одного слова, в качестве S и D должно указываться одно слово.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда первые три цифры слова C не входят в указанный диапазон от 0 до 3. OFF в других случаях.

**Примеры****Передача четырех цифр**

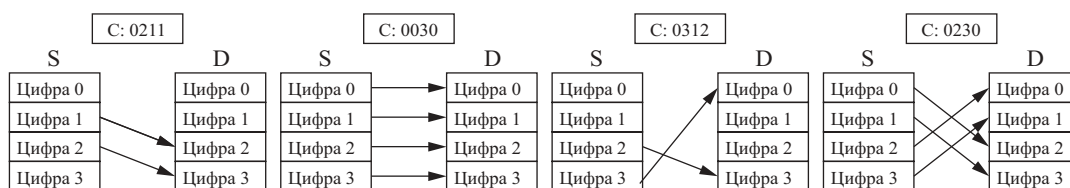
В следующем ниже примере, когда СЮ 000000 находится в состоянии ON, четыре цифры данных записываются из СЮ 0200 в СЮ 0300. Передача начинается с цифры 1 СЮ 0200 и цифры 0 СЮ 0300, в соответствии со значением управляющего слова 0031.



**Примечание:** После прочтения цифры в старшем разряде слова S (цифра 3), команда MOVD(083) возвращается к цифре младшего разряда (цифра 0).

**Примеры с числом С**

На следующем ниже рисунке приводятся примеры передачи данных для различных значений С.

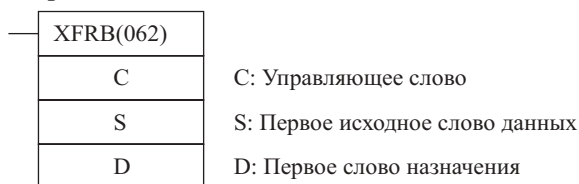


### 3-8-7 Команда передачи множества битов MULTIPLE BIT TRANSFER: XFRB(062)

#### Назначение

По команде XFRB(062) осуществляется передача указанного количества последовательных битов.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	XFRB(062)
	Выполняется один раз для дифференцирования вверх	@ XFRB(062)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

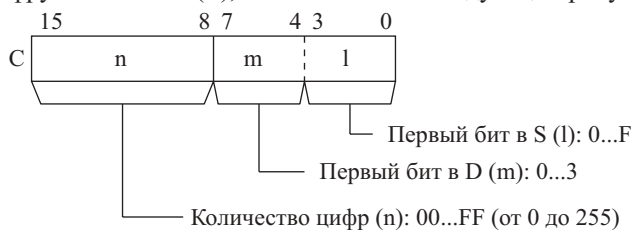
#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

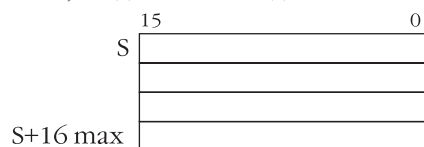
##### C: Управляющее слово

Три цифры младших разрядов слова C указывают на первую исходную цифру (m), вторые три цифры показывают количество цифр, подлежащих передаче (n), затем идут три цифры, показывающие первую цифру назначения (L), как показано на следующем рисунке.



##### S: Первое исходное слово

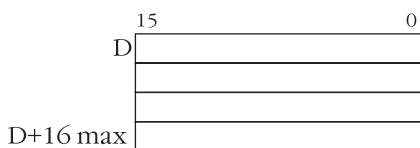
Указывает первое исходное слово. Исходные биты читаются справа налево, последовательно, до слова S+16, когда это необходимо.



**Примечание:** Исходные слова должны находиться в одной области данных.

##### D: Первое слово назначения

Указывает первое слово назначения. Биты записываются справа налево, последовательно, до слова D+16, когда это необходимо.



**Примечание:** Слова назначения должны находиться в одной области данных.

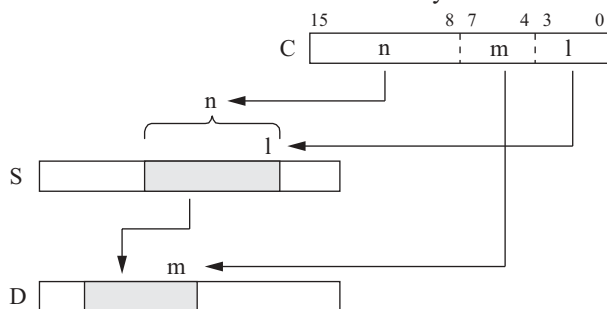
### Спецификации операндов

Область	C	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	Только указанные значения	–	–
Регистры данных	DR0...DR15		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 , IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15		

### Описание

По команде XFRB(062) осуществляется передача до 255 последовательных битов из исходных слов (начиная с бита *l* слова *S*) в слова назначения (начиная с бита *m* слова *D*). Биты, которые в словах назначения не перезаписываются битами исходных слов, не изменяются.

Начальные биты и количество битов указываются в *C*, как показано на следующем ниже рисунке.



Допускается перекрытие исходных слова и слов назначения. Передача данных, которые перекрывают несколько слов, позволяет более эффективно компоновать передаваемые данные в области данных. (Это особенно полезно при обработке данных для управления позиционированием.)

Так как исходные слова и слова назначения могут перекрываться, команда XFRB(062) может объединяться с командой ANDW(034) для смещения *m* битов на *n* пробелов.

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF

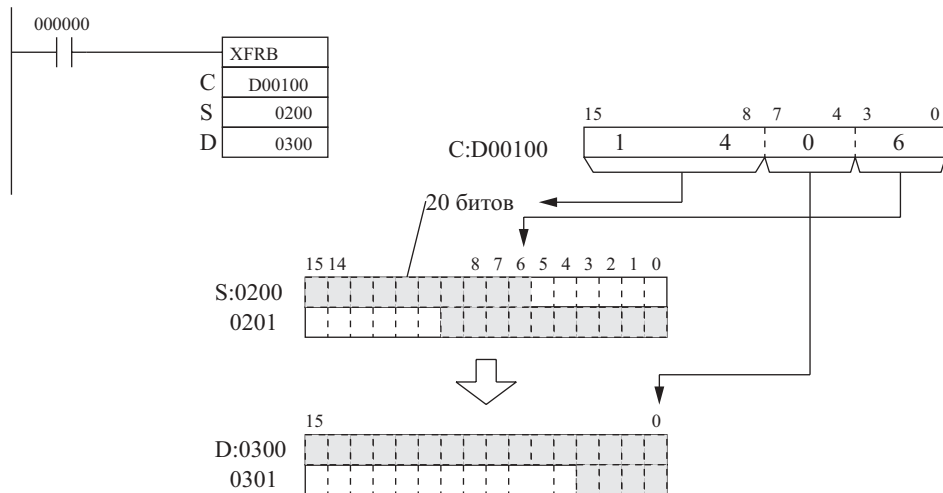


**Меры предосторожности**

При выполнении команды XFRB(062) может передаваться не более 255 битов данных.  
Убедитесь в том, что исходные слова и слова назначения не превышают конца области данных.

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, 20 битов данных, начиная с CIO 020006, записываются в 20 битов, начиная с CIO 030000.



**3-8-8 Команда передачи данных блока BLOCK TRANSFER: XFER(070)**

**Назначение**

По команде XFER(070) осуществляется передача указанного количества последовательных слов.

**Символ релейно-контактной схемы**

XFER(070)	
N	N: Количество слов
S	S: Первое исходное слово данных
D	D: Первое слово назначения

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	XFER(070)
	Выполняется один раз для дифференцирования вверх	@ XFER(070)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

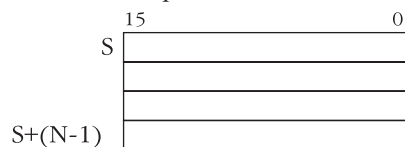
**Операнды**

**N: Количество слов**

Указывает количество слов, подлежащих передаче. Диапазон значений для N – от 0000 до FFFF (от 0 до 65535 в десятичном коде).

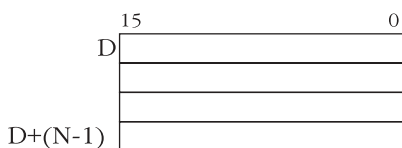
**S: Первое исходное слово**

Указывает первое исходное слово. Слова S и S+(N-1) должны находиться в одной области данных.



**D: Первое слово назначения**

Указывает первое слово назначения. D и D+(N-1) должны находиться в одной области данных.

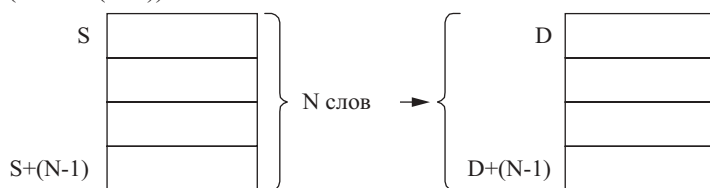


### Спецификации операндов

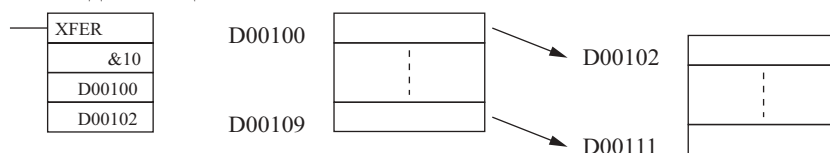
Область	C	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF (двоичные) или &0...&65535	—	—
Регистры данных	DR0...DR15	—	
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15		

### Описание

По команде XFER(070) осуществляется запись N слов, начиная с S (S...S+(N-1)), в N слов, начиная с D (D...D+(N-1)).



Допускается перекрытие исходных слов и слов назначения, поэтому команда XFRB(062) может применяться для смещения слов.



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF

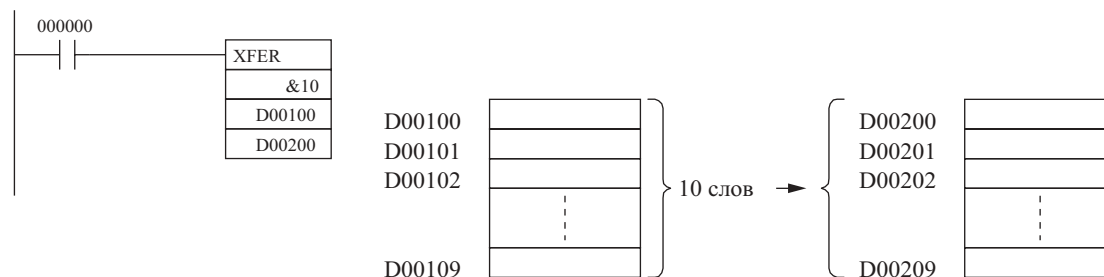
### Меры предосторожности

Убедитесь в том, что исходные слова (S S+(N-1)) и слова назначения (DD+(D-1)) не превышают конца области данных.

В случае, когда осуществляется передача большого количества слов, на выполнение команды требуется некоторое время. В этом случае, передача данных по команде XFER(070) может быть не завершена, если в процессе выполнения команды происходит прерывание подачи питания.

#### Пример

В следующем ниже примере, когда СЮ 000000 находится в состоянии ON, 10 слов данных, начиная с D00100 до D00109, записываются в 10 слов, начиная с D00200 до D00209.

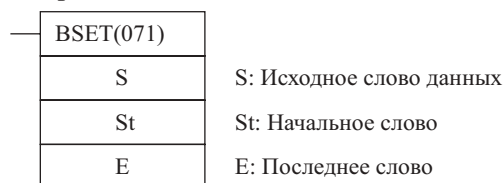


#### 3-8-9 Команда задания блока BLOCK SET: BSET(071)

##### Назначение

По команде BSET(071) осуществляется запись одного и того же слова в диапазон последовательных слов.

##### Символ релейно-контактной схемы



##### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	BSET(071)
	Выполняется один раз для дифференцирования вверх	@ BSET(071)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

##### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

##### Операнды

###### S: Исходное слово

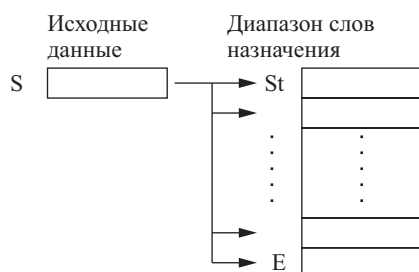
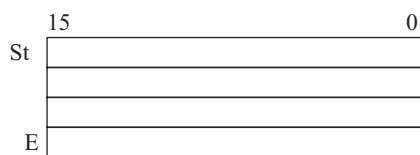
Указывает исходные данные или слово, содержащее исходные данные.

###### St: Начальное слово

Указывает первое слово в диапазоне слов назначения.

###### E: последнее слово

Указывает последнее слово в диапазоне слов назначения.



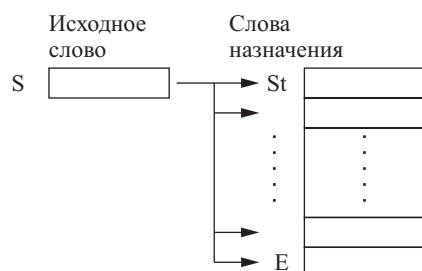
**Примечание:** Слова *St* и *E* должны находиться в одной области данных.

### Спецификации операндов

Область	S	St	E
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A448...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0C)		
Константы	#0000...#FFFF (двоичные)	-	
Регистры данных	DR0...DR15	-	
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 , IR15 DR0...DR15, IR0...IR15 ,IR0+(++)..., IR15+(++) ,-(-)IR0...,(-(-)IR15		

### Описание

По команде BSET(071) производится запись одного исходного слова (S) во все слова назначения от St до E.



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, если St больше E. OFF в других случаях.

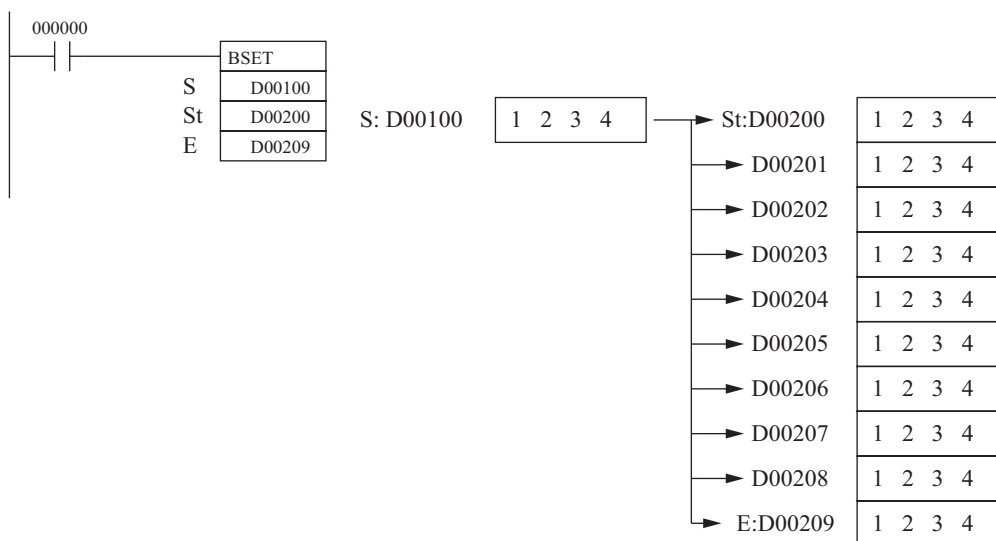
**Меры предосторожности**

Убедитесь в том, что начальное слово (St) и последнее слова (E) находятся в одной области данных, и выполняется соотношение St E.

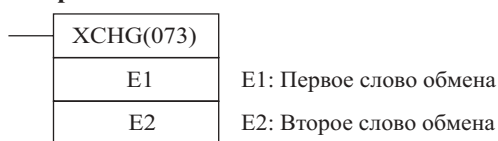
В случае, когда запись исходного слова осуществляется в большое количество слов назначения, на выполнение команды требуется некоторое время. В этом случае, передача данных по команде BSET(071) может быть не завершена, если в процессе выполнения команды происходит прерывание подачи питания.

**Пример**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, исходные данные в D00100 записываются в слова назначения, начиная с D00200 до D00209.

**3-8-10 Команда обмена данными DATA EXCHANGE: XCHG(073)****Назначение**

По команде XCHG(073) осуществляется обмен данными между двумя указанными словами.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	XCHG(073)
	Выполняется один раз для дифференцирования вверх	@ XCHG(073)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

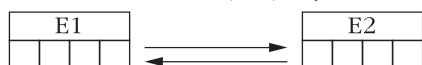
**Спецификации операндов**

Область	E1	E2
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	

Область	E1	E2
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A448...A959	
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#0000...#FFFF (двоичные)	
Регистры данных	DR0...DR15	
Индексные регистры	-	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 , IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,(-)IR15	

**Описание**

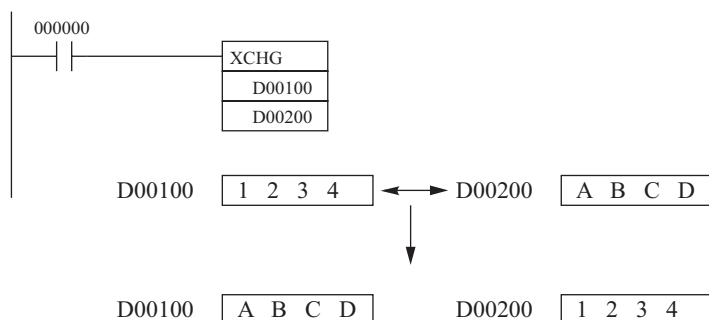
По команде XCHG(073) осуществляется обмен данными между двумя указанными словами.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	OFF
Флаг отрицательного значения	N	OFF

**Пример**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, содержание D00100 обменивается с содержанием D00200.

**3-8-11 Команда обмена двойными данными DOUBLE DATA EXCHANGE: XCGL(562)****Назначение**

По команде XCGL(562) осуществляется обмен содержания пары последовательных слов с другой парой последовательных слов.

## Символ релейно-контактной схемы

XCGL(562)	
E1	E1: Первое слово обмена
E2	E2: Второе слово обмена

## Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	XCGL(562)
	Выполняется один раз для дифференцирования вверх	@ XCGL(562)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

## Применяемые области программы

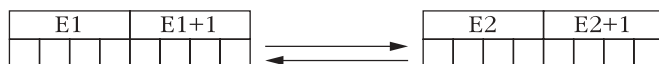
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

## Спецификации операндов

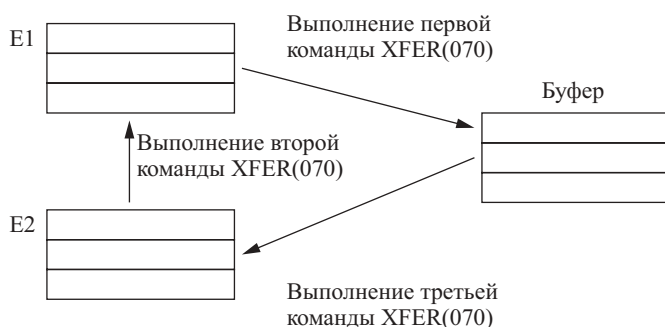
Область	E1	E2
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A448...A958	
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	—	—
Регистры данных	—	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)..., IR15+(++) ,(- -)IR0...,(- -)IR15	

## Описание

По команде XCGL(562) осуществляется обмен содержанием слов E1+1 и E1 с содержанием слов E2+1 и E2.



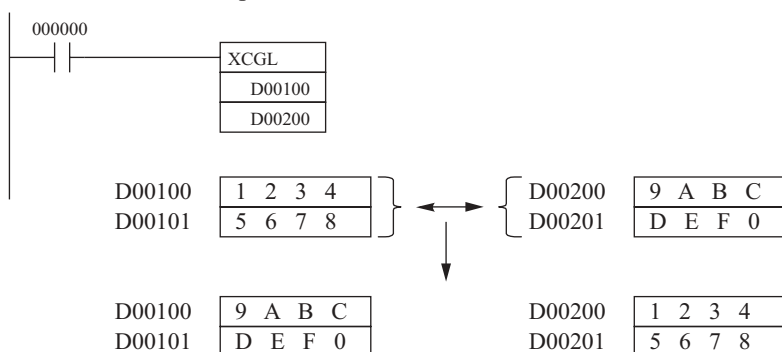
Для обмена содержанием 3-х и более слов, используйте команду XFER(070) для передачи слов в третий набор слов (буфер), как показано на следующем ниже рисунке.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	OFF
Флаг отрицательного значения	N	OFF

**Пример**

В следующем ниже примере, когда СЮ 000000 находится в состоянии ON, содержание D00100 и D00101 обменивается с содержанием D00200 и D00201.

**3-8-12 Команда распределения одного слова SINGLE WORD DISTRIBUTE: DIST(080)****Назначение**

По команде DIST(080) осуществляется передача исходного слова в слово назначения, вычисленное путем прибавления смещения к базовому адресу.

**Символ релейно-контактной схемы**

DIST(080)	
S	S: Исходное слово
Bs	Bs: Базовый адрес назначения
Of	Of: Смещение

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	DIST(080)
	Выполняется один раз для дифференцирования вверх	@ DIST(080)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

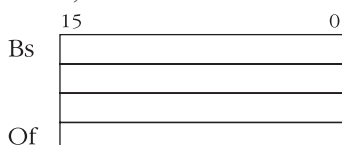


**Операнды****Bs: Базовый адрес назначения**

Указывает базовый адрес назначения. Для вычисления слова назначения к данному адресу прибавляется смещение.

**Of: Смещение**

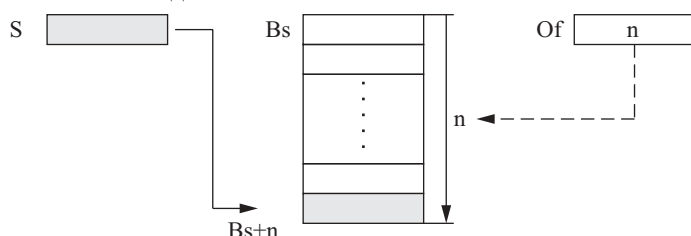
Для вычисления истинного слова назначения к базовому адресу прибавляется данное смещение. Значение смещения может изменяться от 0000 до FFFF (от 0 до 65535 в десятичном коде). Тем не менее, оба значения Bs и Bs+Of должны находиться в одной области данных.

**Спецификации операндов**

Область	S	Bs	Of
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	A000...A959
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0C)		
Константы	#0000...#FFFF (двоичные)	–	#0000...#FFFF (двоичные) или от &0 до &65535
Регистры данных	DR0...DR15	–	DR0...DR15
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047, IR0 -2048...+2047, IR15 DR0...DR15, IR0...IR15 ,IR0(++)..., IR15(++) ,(- -)IR0...,-(- -)IR15		

**Описание**

По команде DIST(080) осуществляется запись содержания S в слово назначения, вычисляемое посредством прибавления смещения Of к базовому адресу Bs. Посредством изменения значения Of одна команда DIST(080) может использоваться для распределения одного исходного слова различным словам назначения в области данных.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF

Наименование	Обозначение	Функционирование
Флаг равенства	=	Переводится в состояние ON, когда исходные данные равны 0000. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда старший бит исходных данных равен 1. OFF в других случаях.

**Меры предосторожности**

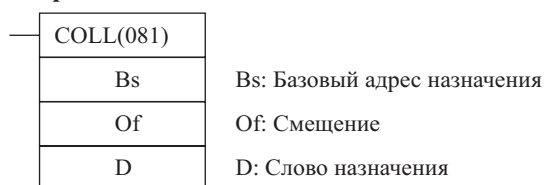
Убедитесь в том, что прибавление смещения не приводит к выходу за границу области данных, т.е. значения Bs и Bs+Of находятся в одной области данных.

**Пример**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, содержание D00100 записывается в D00210 (D00200+10), если содержание D00300 равно 10 (0A в шестнадцатеричном коде). Содержание D00100 может записываться в другие слова путем изменения величины смещения в D00300.

**3-8-13 Команда отбора данных DATA COLLECT: COLL(081)****Назначение**

По команде COLL(081) осуществляется пересылка исходного слова (вычисленного путем прибавления смещения к базовому адресу) в слово назначения.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	COLL(081)
	Выполняется один раз для дифференцирования вверх.	@ COLL(081)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается.	

**Применяемые области программы**

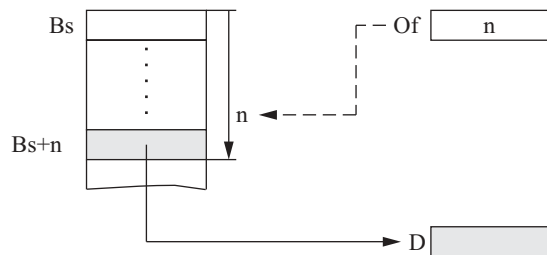
Области блоков программы	Области шагов программы	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****Bs: Базовый адрес источника**

Указывает базовый адрес источника. Для вычисления исходного слова к данному адресу прибавляется смещение.

**Of: Смещение**

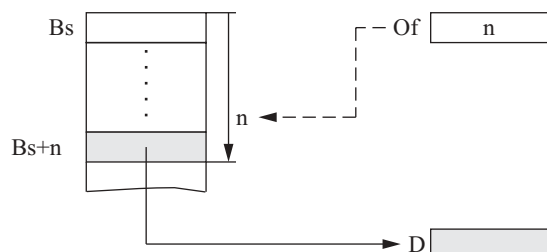
Для вычисления исходного слова к базовому адресу прибавляется данное смещение. Значение смещения может изменяться от 0000 до FFFF (от 0 до 65535 в десятичном коде). Тем не менее, оба значения Bs и Bs+Of должны находиться в одной области данных.

**Спецификации операндов**

Область	Bs	Of	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	–	#0000...#FFFF (двоичные) или от &0 до &65535	–
Регистры данных	–	DR0...DR15	
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0 ,IR15 -2048...+2047 ,IR0 -2048...+2047, IR15 DR0...DR15, IR0...IR15 ,IR0(++),...,IR15(++) ,(- -)IR0...,(- -)IR15		

**Описание**

По команде COLL(081) осуществляется запись содержания исходного слова (вычисляемого посредством прибавления смещения Of к базовому адресу Bs) в слово назначения. Посредством изменения значения смещения Of одна команда COLL(081) может использоваться для отбора данных из различных исходных слов в области данных.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF

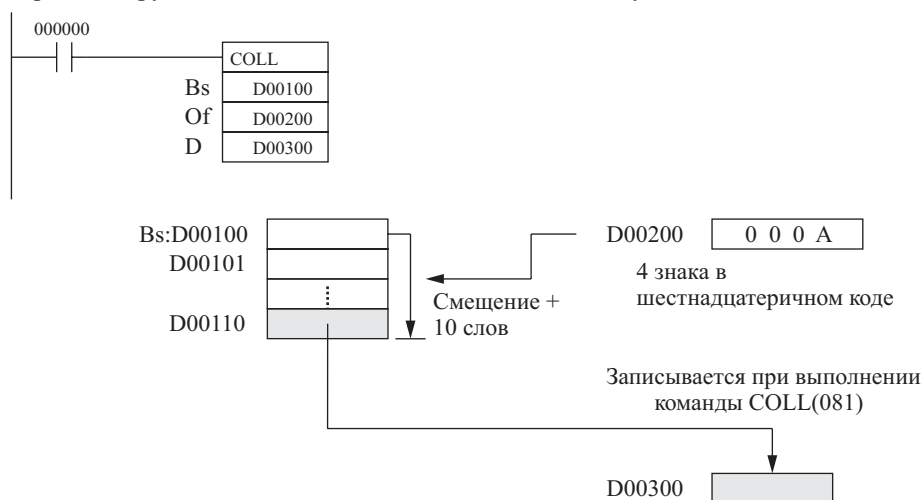
Наименование	Обозначение	Функционирование
Флаг равенства	=	Переводится в состояние ON, когда исходные данные равны 0000. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда старший бит исходных данных равен 1. OFF в других случаях.

### Меры предосторожности

Убедитесь в том, что величина смещения не превышает границы области данных, т.е. значения Bs и Bs+Of находятся в одной области данных.

### Пример

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, содержание D00110 (D00100+10) записывается в D00300, если содержание D00200 равно 10 (0A в шестнадцатеричном коде). Содержание других слов может записываться в D00300 путем изменения величины смещения в D00200.

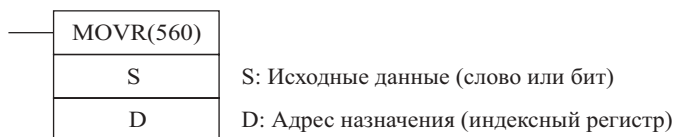


### 3-8-14 Команда перемещения в регистр MOVE TO REGISTER: MOVR(560)

#### Назначение

По команде MOVR(560) осуществляется задание адреса памяти внутреннего ввода/вывода для определенного слова, бита, флага завершения таймера /счетчика в указанном индексном регистре. (Для задания в индексном регистре адреса памяти внутреннего ввода/вывода, принадлежащего текущему значению таймера/счетчика используйте команду MOVRW(561)).

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	MOVR(560)
	Выполняется один раз для дифференцирования вверх.	@ MOVR(560)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

##### D: адрес назначения

Адресом назначения должен быть индексный регистр (IR0...IR15).

### Спецификации операндов

Область	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143 CIO 000000...CIO 614315	–
Рабочая область	W000...W511 W00000...W51115	–
Область удержания бита (Holding bit area)	H000...H511 H00000...H51115	–
Область вспомогательных битов	A000...A447 A448...A959 A00000...A44715 A44800...A95915	–
Область таймера	T0000...T4095 (Флаг завершения)	–
Область счетчика	C0000...C4095 (Флаг завершения)	–
Область DM	D00000...D32767	–
Область EM, не содержащая банков	E00000...E32767	–
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	–
Адреса DM/EM памяти при косвенном указании в двоичном коде	–	–
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–	–
Константы	–	–
Регистры данных	–	–
Индексные регистры	–	IR0...IR15
Косвенная адресация с использованием индексных регистров	–	–

### Описание

По команде MOVR(560) осуществляется поиск адреса памяти внутреннего ввода/вывода (абсолютного адреса) источника и запись этого адреса в D (индексный регистр).



Если в S указывается таймер или счетчик, по команде MOVR(560) производится запись в D адреса памяти, принадлежащего флагу завершения таймера или счетчика. Используйте команду MOVRW(561) для записи в слово D адреса памяти внутреннего ввода/вывода, принадлежащего текущему значению таймера/счетчика.

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	OFF
Флаг отрицательного значения	N	OFF

### Меры предосторожности

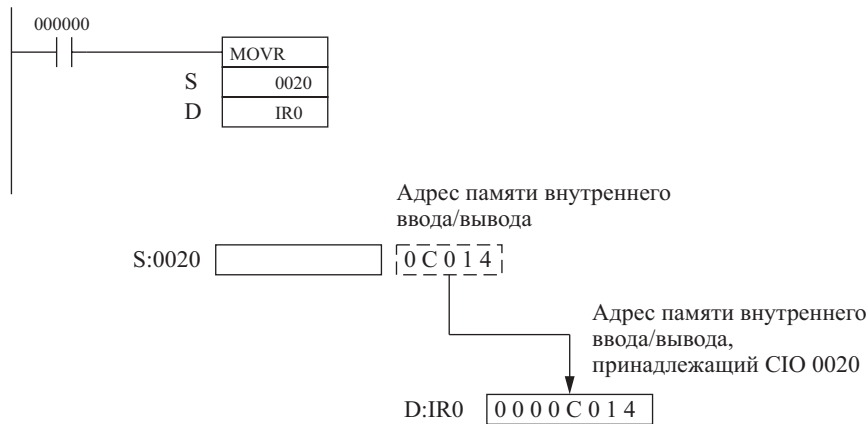
Команда MOVR(560) не может устанавливать адрес памяти внутреннего ввода/вывода для текущего значения таймера/счетчика. Для задания адреса текущего значения таймера/счетчика используйте команду MOVRW(561).

Содержание индексного регистра в задаче прерывания непредсказуемо, пока оно не задано. Поэтому перед использованием индексного регистра в задаче прерывания, непременно с помощью команды MOVR(560) задавайте его содержание.

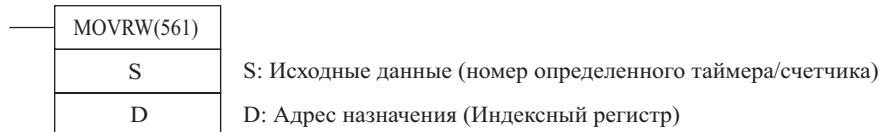
Любое изменение содержания индексного регистра или регистра данных, произведенное в задаче прерывания, не оказывает влияния на содержание регистра в циклической задаче.

**Пример**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, по команде MOVR(560) производится запись адреса памяти внутреннего ввода/вывода, принадлежащего CIO 0020, в индексный регистр IR0.

**3-8-15 Команда записи MOVE TIMER/COUNTER PV TO REGISTER: MOVRW(561)****Назначение**

По команде MOVRW(561) осуществляется запись адреса памяти внутреннего обмена, принадлежащий текущему значению указанного таймера/счетчика, в указанный индексный регистр. (Для задания в индексном регистре адреса слова, бита, флага завершения таймера/счетчика используйте команду MOVR(560)).

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	MOVRW(561)
	Выполняется один раз для дифференцирования вверх	@ MOVRW(561)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****D: адрес назначения**

Адресом назначения должен быть индексный регистр (IR0...IR15).

**Спецификации операндов**

Область	S	D
Область ввода/вывода (область CIO)	—	—
Рабочая область	—	—
Область удержания бита (Holding bit area)	—	—
Область вспомогательных битов	—	—
Область таймера	T0000...T4095 (Текущее значение)	—
Область счетчика	C0000...C4095 (Текущее значение)	—

Область	S	D
Область DM	–	
Область EM, не содержащая банков	–	
Область EM, содержащая банки	–	
Адреса DM/EM памяти при косвенном указании в двоичном коде	–	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–	
Константы	–	
Регистры данных	–	
Индексные регистры	–	IR0...IR15
Косвенная адресация с использованием индексных регистров	–	

**Описание**

По команде MOVR(560) осуществляется поиск адреса памяти внутреннего обмена, принадлежащего текущему значению таймера или счетчика, указанного в S, и запись этого адреса в D (индексный регистр).



По команде MOVR(560) производится запись в D адреса текущего значения таймера или счетчика. Для записи в D адреса флага завершения таймера/счетчика используйте команду MOVR(560).

**Флаги**

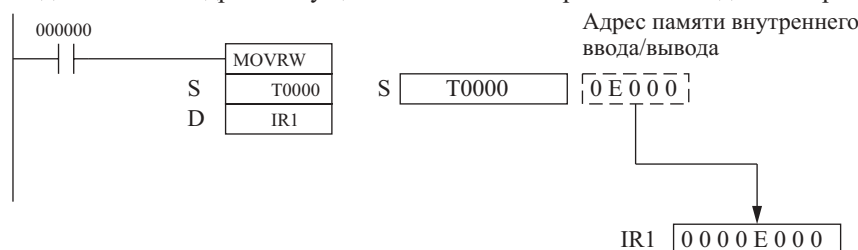
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	OFF
Флаг отрицательного значения	N	OFF

**Меры предосторожности**

Команда MOVRW(561) не может задавать адреса в области данных, т.е. адреса слов, битов, флагов завершения таймеров/счетчиков. Для задания этих адресов используйте команду MOVR(560).

**Пример**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, по команде MOVRW(561) производится запись адреса текущего значения таймера T0000 в индексный регистр IR1.



## 3-9 Команды смещения данных

В настоящем разделе приводится описание команд, используемых для смещения данных в словах или между словами, а также в различных количествах и направлениях.

Команда	Мнемоническое изображение	Функциональный код	Страница
SHIFT REGISTER	SFT	010	225
REVERSIBLE SHIFT REGISTER	SFTR	084	227
ASYNCHRONOUS SHIFT REGISTER	ASFT	017	229
WORD SHIFT	WSFT	016	232
ARITHMETIC SHIFT LEFT	ASL	025	233
DOUBLE SHIFT LEFT	ASLL	570	235
ARITHMETIC SHIFT RIGHT	ASR	026	237
DOUBLE SHIFT RIGHT	ASRL	571	238
ROTATE LEFT	ROL	027	240
DOUBLE ROTATE LEFT	ROLL	572	242
ROTATE LEFT WITHOUT CARRY	RLNC	574	243
DOUBLE ROTATE LEFT WITHOUT CARRY	RLNL	576	245
ROTATE RIGHT	ROR	028	247
DOUBLE ROTATE RIGHT	RORL	573	248
ROTATE RIGHT WITHOUT CARRY	RRNC	575	250
DOUBLE ROTATE RIGHT WITHOUT CARRY	RRNL	577	251
ONE DIGIT SHIFT LEFT	SLD	074	253
ONE DIGIT SHIFT RIGHT	SRD	075	255
SHIFT N-BIT DATA LEFT	NSFL	578	256
SHIFT N-BIT DATA RIGHT	NSFR	579	258
SHIFT N-BITS LEFT	NASL	580	260
DOUBLE SHIFT N-BITS LEFT	NSLL	582	263
SHIFT N-BITS RIGHT	NASR	581	266
DOUBLE SHIFT N-BITS RIGHT	NSRL	583	269

## 3-9-1 Команда сдвига регистра SHIFT REGISTER: SFT(010)

**Назначение**

По команде SFT(010) осуществляется смещение регистра.

**Символ релейно-контактной схемы**

Ввод данных	—	SFT(010)	
Ввод смещения	—	St	St: Начальное слово
Ввод перезапуска	—	E	E: Завершающее слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	SFT(010)
	Выполняется один раз для дифференцирования вверх	Не поддерживается.
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не используется	Да	Да	Да

**Примечание:** St и E должны находиться в одной области данных.

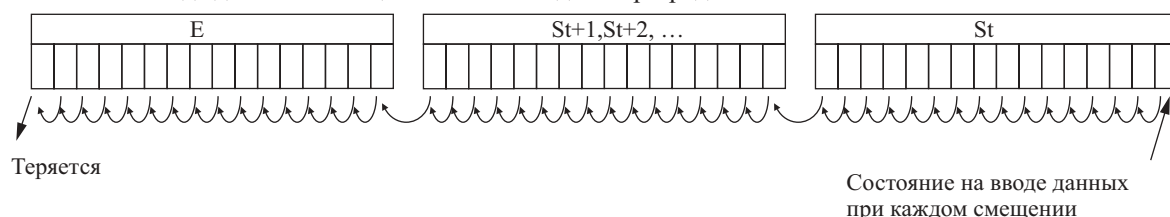


### Спецификации операндов

Область	St	E
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A448...A959	
Область таймера	-	
Область счетчика	-	
Область DM	-	
Область EM, не содержащая банков	-	
Область EM, содержащая банки	-	
Адреса DM/EM памяти при косвенном указании в двоичном коде	-	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	-	
Константы	-	
Регистры данных	-	
Индексные регистры	-	
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-(-)IR15	

### Описание

Когда условие выполнения на вводе смещения переходит из состояния OFF в состояние ON, все данные от St до E смещаются влево на один бит (от бита младшего разряда до бита старшего разряда), а состояние ON/OFF на вводе данных помещается в бит младшего разряда.



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда косвенно указанный адрес регистра IR для S и E находится за пределами областей данных CIO, AR, HR, или WR. OFF в других случаях.

### Меры предосторожности

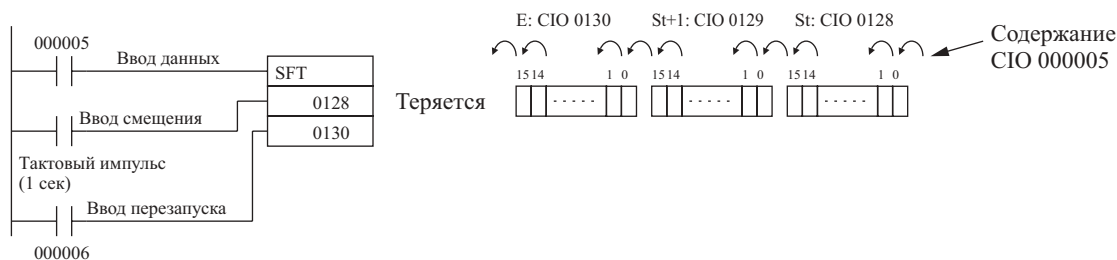
Бит данных, смещенный за пределы индексного регистра, в дальнейшем не учитывается.

Когда ввод перезапуска переводится в состояние ON, все биты индексного регистра, начиная с указанного младшего слова (St) и заканчивая указанным старшим словом (E), переустанавливаются (т.е. сбрасываются в значение, равное нулю). Ввод переустановки обладает приоритетом по отношению к другим вводам. Значение St должно быть меньше или равно значению E, однако, даже когда величина St больше E, ошибка не определяется и одно слово данных в St будет смещено.

Когда St и E указываются косвенным способом с использованием индексных регистров и действительные адреса памяти ввода/вывода находятся вне областей памяти для данных, определяется ошибка и флаг ошибки переводится в состояние ON.

### Примеры

В следующем ниже примере, показано смещение регистра с использованием слов CIO 0128...CIO 0130. Тактовый импульс (1 сек) используется таким образом, что условие выполнения, производимое в CIO 000005, каждую секунду смещается в регистр из 3-х слов между CIO 012800 и CIO 013015.

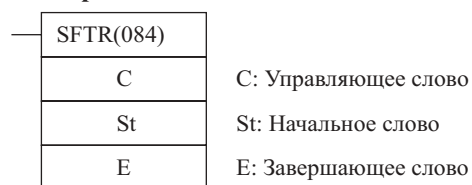


### 3-9-2 Команда реверсивного смещения регистра REVERSIBLE SHIFT REGISTER: SFTR(084)

#### Назначение

По команде SFTR(084) создается регистр смещения, осуществляющий сдвиг данных вправо или влево.

#### Символ релейно-контактной схемы



#### Модификации

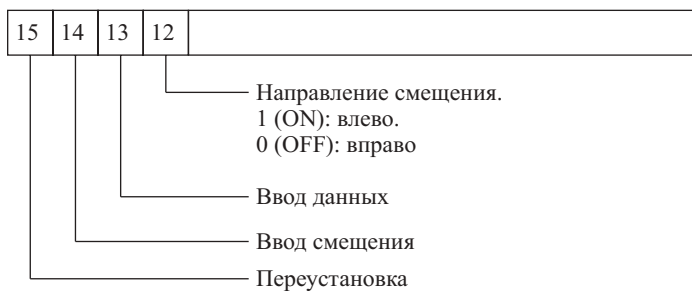
Модификации	Выполняется в каждом цикле для состояния ON.	SFTR(084)
	Выполняется один раз для дифференцирования вверх	@ SFTR(084)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

##### C: Управляющее слово



**Примечание:** St и E должны находиться в одной области данных.

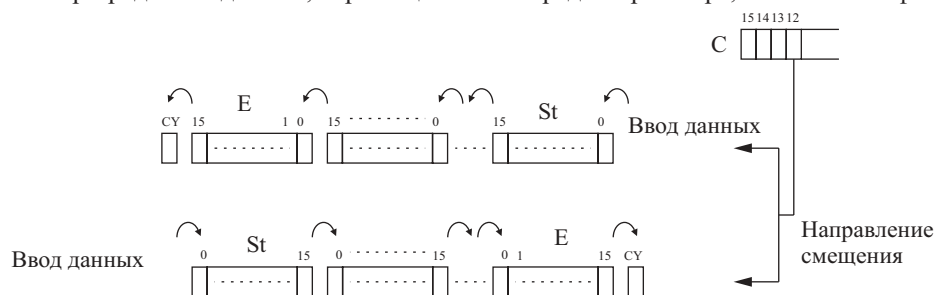
#### Спецификации операндов

Область	C	St	E
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		

Область	C	St	E
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	–		
Регистры данных	DR0...DR15	–	
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...(- -)IR15		

### Описание

Когда условие выполнения на вводе смещения (бит 14 слова C) изменяется из состояния OFF в состояние ON, все данные от St до E перемещаются в заданном направлении (определяемом состоянием бита 12 слова C) на 1 бит. Одновременно состояние ON/OFF на вводе данных записывается в бит младшего или старшего разряда. Бит данных, перемещаемый за пределы регистра, заносится во флаг переноса (CY).



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда St больше E. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда 1 заносится во флаг переноса. Переводится в состояние OFF, когда 0 заносится во флаг переноса. OFF, когда ввод переноса устанавливается в состояние 1.

### Меры предосторожности

Указанные выше операции смещения применяются, когда бит переустановки (бит 15 слова C) устанавливается в состояние OFF.

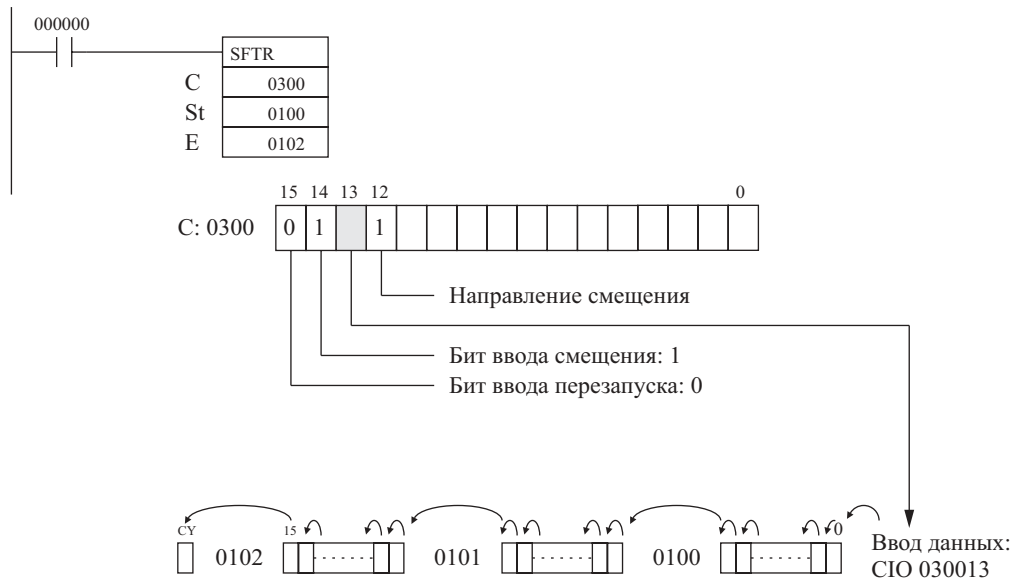
Когда бит переустановки (бит 15 слова C) переводится в состояние ON, все биты от St до E переустанавливаются (т.е. сбрасываются в значение 0).

В случае, когда St больше E, определяется ошибка, и флаг ошибки переводится в состояние ON.

### Примеры

#### Смещение данных

Если ввод смещения CIO 030014 переводится в состояние ON в то время, когда CIO 000000 находится в состоянии ON, а бит переустановки CIO 030015 - в состоянии OFF, слова от CIO 0100 до CIO 0102 смещаются на один бит в направлении, определяемом состоянием CIO 030012 (т.е. 1: вправо). В этом случае содержание бита ввода CIO 030013 заносится в бит младшего разряда CIO 010000. Содержание CIO 010215 заносится во флаг переноса (CY).



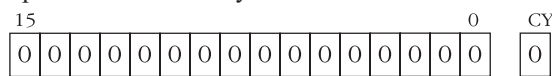
**Переустановка данных**

Когда CIO 030014 и CIO 000000 одновременно находятся в состоянии ON, и бит перезапуска CIO 030015 переводится в состояние ON, слово от CIO 0100 до CIO 0102 и флаг переноса переустанавливаются в значение OFF.

**Управление данными**

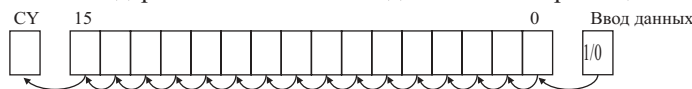
**Переустановка данных**

Когда бит переустановки переводится в состояние ON (бит 15 слова C), все биты от St до E, а также флаг переноса (CY), устанавливаются в значение, равное нулю. При этом никакие данные приниматься не могут.



**Смещение данных влево (начиная с младшего, до старшего бита включительно)**

Когда бит ввода смещения (бит 14 слова C) переводится в состояние ON, содержание бита ввода (бит 13 слова C) смещается в бит 00 начального слова, а все последующие биты смещаются на один бит влево. Содержание бита 15 последнего слова перемещается во флаг переноса (CY).



**Смещение данных вправо (начиная со старшего, до младшего бита включительно)**

Когда бит ввода смещения (бит 14 слова C) переводится в состояние ON, содержание бита ввода (бит 13 слова C) смещается в бит 15 завершающего слова, а все последующие биты смещаются на один бит вправо. Состояние бита 00 начального слова перемещается во флаг переноса (CY).



3-9-3 Команда ASYNCHRONOUS SHIFT REGISTER: ASFT(017)

**Назначение**

По команде SFTR(017) осуществляется смещение всех значащих (ненулевых) слов данных в заданном диапазоне слов либо в направлении St, либо в направлении E, замещая слова данных, содержащие 0000 в шестнадцатеричном коде.

**Символ релейно-контактной схемы**

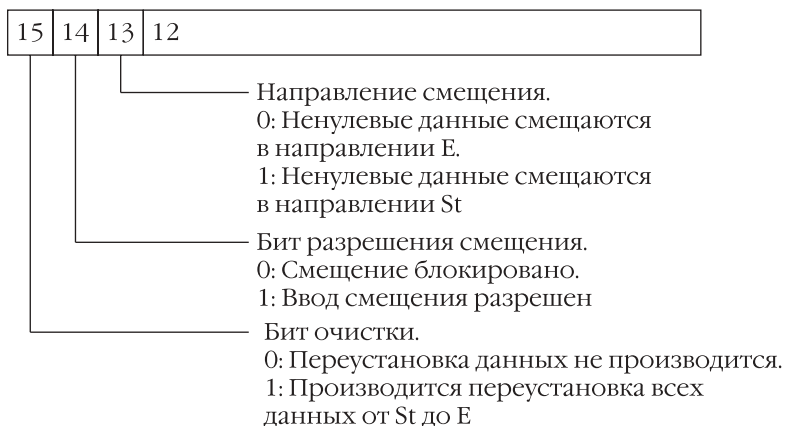
ASFT(017)	
C	C: Управляющее слово
St	St: Начальное слово
E	E: Завершающее слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	ASFT(017)
	Выполняется один раз для дифференцирования вверх	@ ASFT(017)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****C: Управляющее слово**

**Примечание:** Слова St и E должны находиться в одной области данных.

**Спецификации операндов**

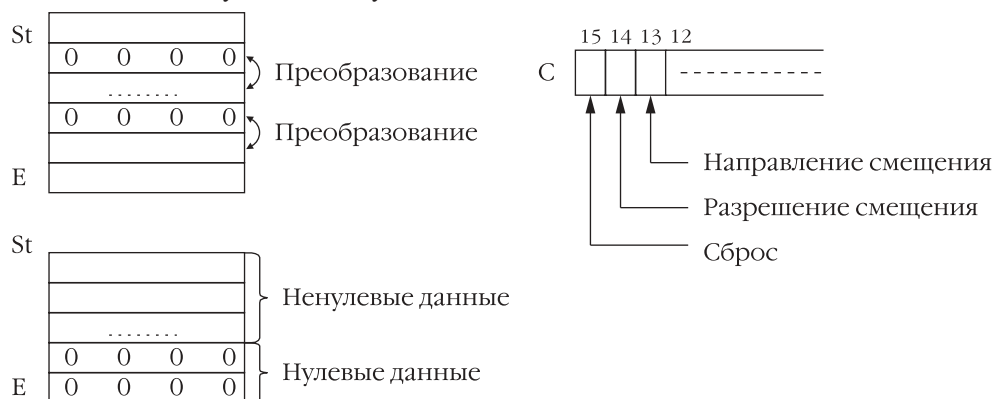
Область	C	St	E
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	-		
Регистры данных	DR0...DR15	-	
Индексные регистры	-		

Область	C	St	E
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-(-)IR15		

### Описание

Когда бит разрешения смещения (бит 14 слова C) переводится в состояние ON, все ненулевые слова в диапазоне от St до E перемещаются на одно слово в направлении, определяемом состоянием бита 13 в слове C, если при этом слово, находящееся в направлении смещения, содержит все нули.

Если команда ASFT(017) повторяется достаточное количество раз, все слова, содержащие нулевые данные, заменяются словами не содержащими нулевые данные. Это приводит к тому, что все данные между St и E делятся на нулевые и ненулевые данные.



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда St больше E. OFF в других случаях.

### Меры предосторожности

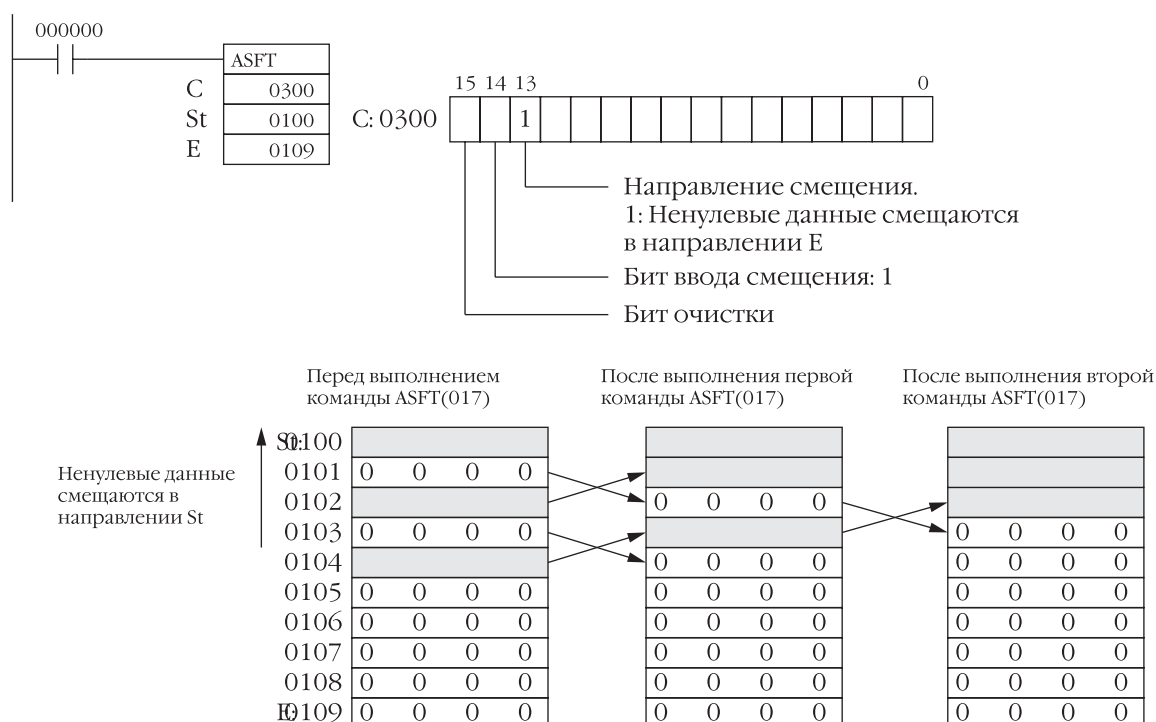
Когда бит переустановки (бит 15 слова C) переводится в состояние ON, все биты от St до E переустанавливаются (т.е. сбрасываются в значение 0). Флаг переустановки обладает приоритетом по отношению к биту разрешения переноса (бит 14 слова C).

В случае, когда St больше E, определяется ошибка, и флаг ошибки переводится в состояние ON.

### Примеры

#### Смещение данных

Когда бит разрешения переноса CIO 030014 переводится из состояния OFF в состояние ON, а бит CIO 000000 одновременно находится в состоянии ON, все слова с ненулевыми данными от CIO 0100 до CIO 0109 сдвигаются в направлении, определяемом состоянием бита направления смещения CIO 030013 (т.е. 1: в направлении St), если при этом слово, находящееся слева содержит все нули.

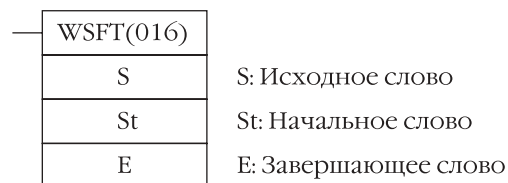


### 3-9-4 Команда смещения слова WORD SHIFT : WSFT(016)

#### Назначение

По команде WSFT(016) осуществляется смещение данных между St и E в единицах, равных одному слову.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	WSFT(016)
	Выполняется один раз для дифференцирования вверх	@ WSFT(016)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Примечание:** St и E должны находиться в одной области данных.

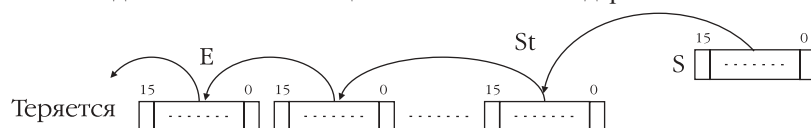
#### Спецификации операндов

Область	S	St	E
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		

Область	S	St	E
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF (двоичные)	-	
Регистры данных	DR0...DR15	-	
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,(-)IR15		

**Описание**

По команде WSFT(016) производится смещение данных от St к E в единицах, равных одному слову. Данные исходного слова S помещаются в слово St. Содержание слова E теряется.



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда St больше E. OFF в других случаях.

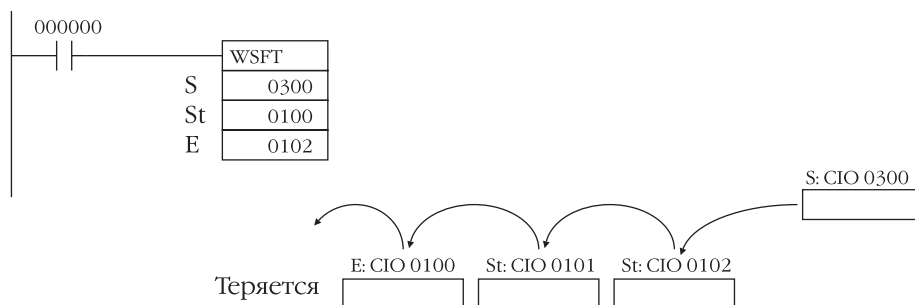
**Меры предосторожности**

В случае, когда St больше E, определяется ошибка, и флаг ошибки переводится в состояние ON.

**Примечание:** При смещении большого количества данных время выполнения команды достаточно велико. Предпринимайте меры по предотвращению отключения питания при выполнении команды WSFT(016) во избежание прерывания операции смещения данных до ее полного завершения.

**Примеры**

Когда бит CIO 000000 переводится в состояние ON, все данные от CIO 0100 до CIO 0102 сдвигаются на одно слово в направлении E. Содержание CIO 0300 сохраняется в CIO 0100, а содержание CIO 0102 теряется.

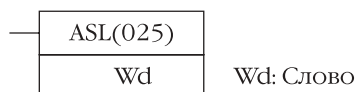


3-9-5 Команда арифметического смещения влево ARITHMETIC SHIFT LEFT: ASL(025)

**Назначение**

По команде WSFT(016) осуществляется смещение содержания слова Wd на один бит влево.



**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	ASL(025)
	Выполняется один раз для дифференцирования вверх	@ ASL(025)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

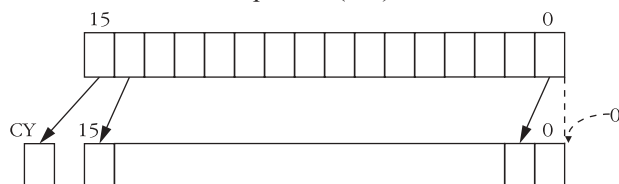
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143
Рабочая область	W000...W511
Область удержания бита (Holding bit area)	H000...H511
Область вспомогательных битов	A448...A959
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область DM	D00000...D32767
Область EM, не содержащая банков	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)
Константы	—
Регистры данных	DR0...DR15
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15

**Описание**

По команде ASL(025) осуществляется смещение слова Wd на один бит влево (от бита младшего разряда до бита старшего разряда). В бит младшего разряда заносится «0», а данные бита старшего разряда перемещаются во Флаг переноса (CY).

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF.
Флаг равенства	=	Переводится в состояние ON, когда результатом сдвига является 0. OFF в других случаях.

Наименование	Обозначение	Функционирование
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении смещения во Флаг переноса (CY) заносится 1. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда при выполнении смещения в бите старшего разряда появляется единица. OFF в других случаях.

### Меры предосторожности

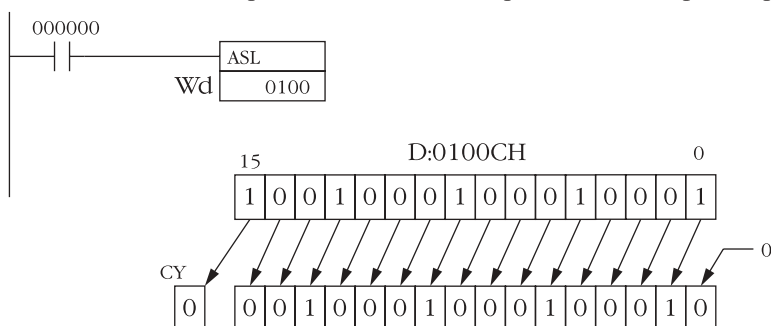
После выполнения команды ASL(025) флаг ошибки переводится в состояние OFF.

Если в результате смещения данных в слове Wd содержится нулевое значение, флаг равенства переводится в состояние ON.

Когда в результате смещения данных в бит старшего разряда слова Wd заносится 1, флаг отрицательного значения переводится в состояние ON.

### Примеры

Когда бит CIO 000000 переводится в состояние ON, CIO 0100 смещается на один бит влево. В CIO 010000 заносится «0», а содержание CIO 010115 перемещается во флаг переноса (CY).

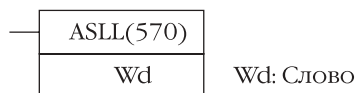


### 3-9-6 Команда смещения двойных данных DOUBLE SHIFT LEFT: ASLL(570)

#### Назначение

По команде ASLL(570) осуществляется смещение содержания слов Wd и Wd+1 на один бит влево.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	ASLL(570)
	Выполняется один раз для дифференцирования вверх.	@ ASLL(570)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается.	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

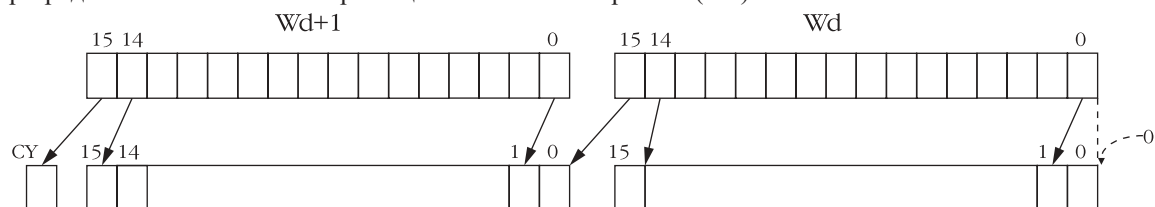
#### Спецификации операндов

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142
Рабочая область	W000...W510
Область удержания бита (Holding bit area)	H000...H510
Область вспомогательных битов	A448...A958
Область таймера	T0000...T4094
Область счетчика	C0000...C4094

Область	Wd
Область DM	D00000...D32766
Область EM, не содержащая банков	E00000...E32766
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)
Константы	–
Регистры данных	–
Индексные регистры	–
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048+2047,IR0 -2048...+2047,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-(-)IR15

### Описание

По команде ASLL(570) производится смещение слов Wd и Wd+1 на один бит влево (от бита младшего разряда до бита старшего разряда). В бит младшего разряда слова Wd заносится «0», а данные битов старших разрядов слов Wd и Wd+1 перемещаются во Флаг переноса (CY).



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF.
Флаг равенства	=	Переводится в состояние ON, когда результатом сдвига является 0. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении смещения во Флаг переноса (CY) заносится 1. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда при выполнении смещения в бите старшего разряда появляется единица. OFF в других случаях.

### Меры предосторожности

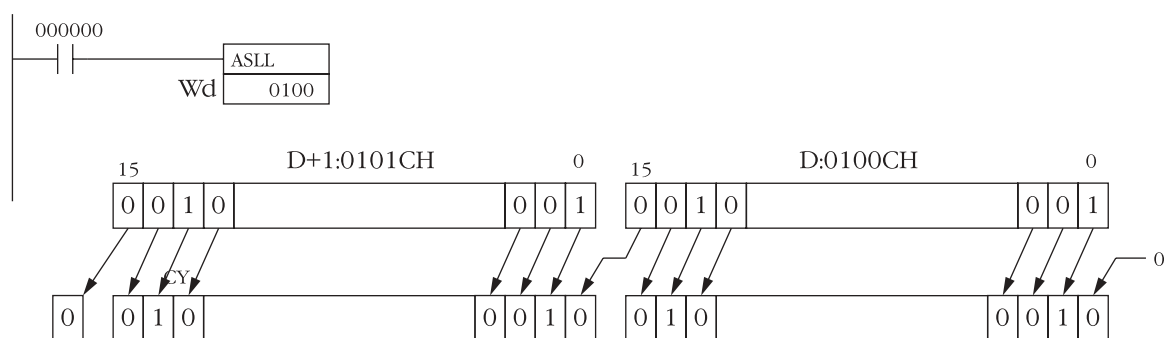
После выполнения команды ASLL(570) флаг ошибки переводится в состояние OFF.

Если в результате смещения данных в словах Wd и Wd+1 содержатся нулевые значения, флаг равенства переводится в состояние ON.

Когда в результате смещения данных в бит старшего разряда слова Wd+1 заносится 1, флаг отрицательного значения переводится в состояние ON.

### Примеры

Когда бит CIO 000000 переводится в состояние ON, CIO 0100 и CIO 0101 смещаются на один бит влево. В CIO 010000 заносится «0», а содержание CIO 010115 перемещается во флаг переноса (CY).

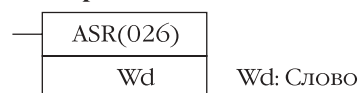


### 3-9-7 Команда арифметического сдвига вправо ARITHMETIC SHIFT RIGHT: ASR(026)

#### Назначение

По команде ASR(026) осуществляется сдвиг содержания слова Wd на один бит вправо.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	ASR(026)
	Выполняется один раз для дифференцирования вверх	@ ASR(026)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

#### Применяемые области программы

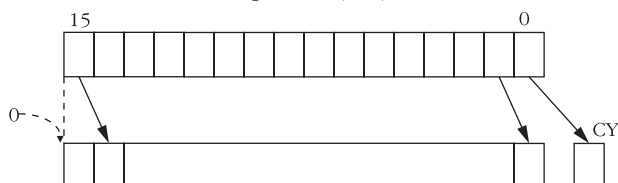
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Спецификации операндов

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143
Рабочая область	W000...W511
Область удержания бита (Holding bit area)	H000...H511
Область вспомогательных битов	A448...A959
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область DM	D00000...D32767
Область EM, не содержащая банков	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)
Константы	-
Регистры данных	DR0...DR15i
Индексные регистры	-
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048+2047,IR0-2048...+2047,IR15 DR0...DR15,IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-)IR15

**Описание**

По команде ASR(026) осуществляется смещение слова Wd на один бит вправо (от бита старшего разряда до бита младшего разряда). В бит старшего разряда заносится «0», а данные бита младшего разряда перемещаются во Флаг переноса (CY).

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF.
Флаг равенства	=	Переводится в состояние ON, когда результатом сдвига является 0. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении смещения во Флаг переноса (CY) заносится 1. OFF в других случаях.
Флаг отрицательного значения	N	OFF

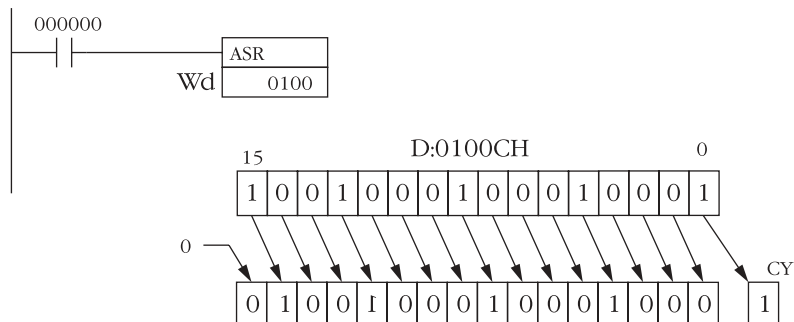
**Меры предосторожности**

После выполнения команды ASR(026) флаг ошибки и флаг отрицательного значения переводятся в состояние OFF.

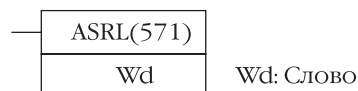
Если в результате смещения данных в слове Wd содержится нулевое значение, флаг равенства переводится в состояние ON.

**Примеры**

Когда бит CIO 000000 переводится в состояние ON, CIO 0100 смещается на один бит вправо. В CIO 010015 заносится «0», а содержание CIO 010000 перемещается во флаг переноса (CY).

**3-9-8 Команда смещения двойных данных вправо DOUBLE SHIFT RIGHT: ASRL(571)****Назначение**

По команде ASRL(571) осуществляется смещение содержания слов Wd и Wd+1 на один бит вправо.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	ASRL(571)
	Выполняется один раз для дифференцирования вверх	@ ASRL(571)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.		Не поддерживается

**Применяемые области программы**

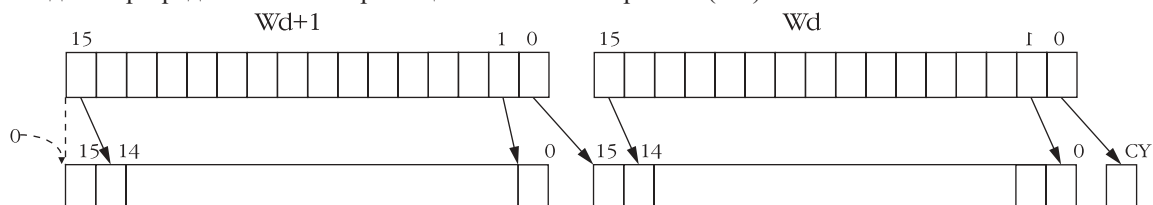
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142
Рабочая область	W000...W510
Область удержания бита (Holding bit area)	H000...H510
Область вспомогательных битов	A448...A958
Область таймера	T0000...T4094
Область счетчика	C0000...C4094
Область DM	D00000...D32766
Область EM, не содержащая банков	E00000...E32766
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)
Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15

**Описание**

По команде ASRL(571) осуществляется смещение слов Wd и Wd+1 на один бит вправо (от бита старшего разряда до бита младшего разряда). В бит старшего разряда слова Wd+1 заносится «0», а содержание бита младшего разряда слова Wd перемещается во Флаг переноса (CY).

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF.
Флаг равенства	=	Переводится в состояние ON, когда результатом сдвига является 0. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении смещения во Флаг переноса (CY) заносится 1. OFF в других случаях.
Флаг отрицательного значения	N	OFF

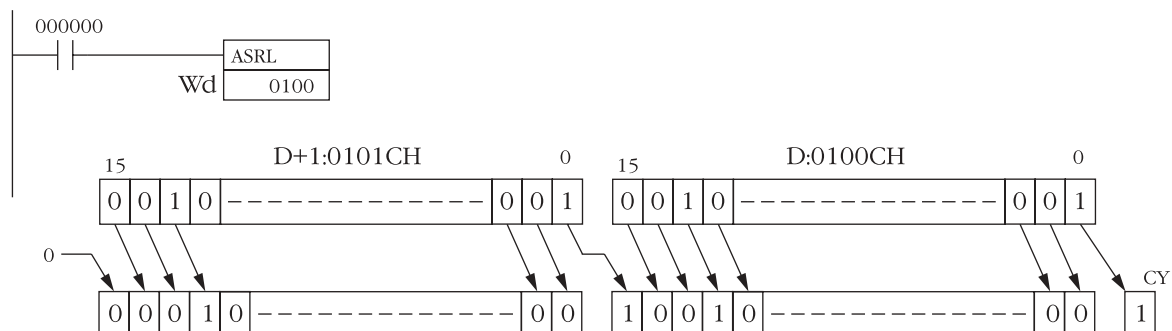
**Меры предосторожности**

После выполнения команды ASLL(570) флаг ошибки и флаг отрицательного значения переводятся в состояние OFF.

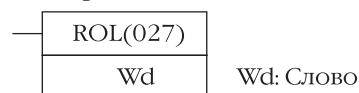
Если в результате смещения данных в словах Wd и Wd+1 содержатся нулевые значения, флаг равенства переводится в состояние ON.

**Примеры**

Когда бит CIO 000000 переводится в состояние ON, слова CIO 0100 и CIO 0101 смещаются на один бит вправо. В CIO 010115 заносится «0», а содержание CIO 010000 перемещается во флаг переноса (CY).

**3-9-9 Команда вращения влево ROTATE LEFT: ROL(027)****Назначение**

По команде ROL(027) осуществляется смещение всех битов слова Wd, включая флаг переноса (CY), на один бит влево.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	ROL(027)
	Выполняется один раз для дифференцирования вверх	@ ROL(027)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается.	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

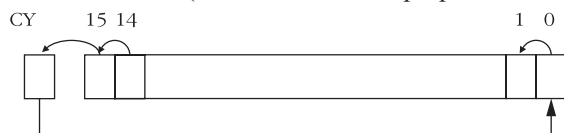
**Спецификации операндов**

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143
Рабочая область	W000...W511
Область удержания бита (Holding bit area)	H000...H511
Область вспомогательных битов	A448...A959
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область DM	D00000...D32767
Область EM, не содержащая банков	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)
Константы	—
Регистры данных	DR0...DR15
Индексные регистры	—

Область	Wd
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(++) ,-( -)IR0...,-( -)IR15

### Описание

По команде ROL(027) осуществляется смещение всех битов слова Wd, включая флаг переноса (CY), на один бит влево (от бита младшего разряда до бита старшего разряда).



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF.
Флаг равенства	=	Переводится в состояние ON, когда результатом сдвига является 0. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении смещения во Флаг переноса (CY) заносится 1. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда при выполнении смещения в старший бит заносится 1. OFF в других случаях.

### Меры предосторожности

После выполнения команды ROL(027) флаг ошибки переводится в состояние OFF.

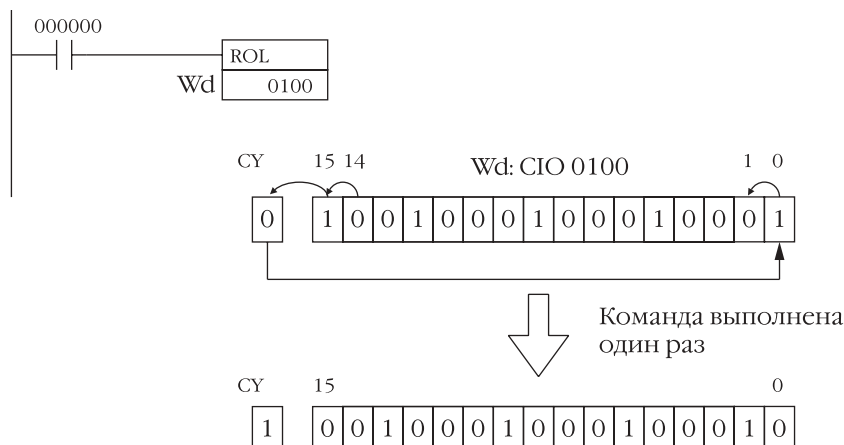
Если в результате смещения данных в слове Wd содержится нулевое значение, флаг равенства переводится в состояние ON.

Если в результате выполнения смещения данных содержанием бита старшего разряда слова Wd становится 1, Флаг отрицательного значения переводится в состояние ON.

**Примечание:** Используя команды Set Carry (STC(040)) и Clear Carry (CLC(041)) можно установить флаг переноса в состояние 1 или 0 непосредственно перед выполнением команды ROL(027).

### Примеры

Когда бит CIO 000000 переводится в состояние ON, слово CIO 0100 и флаг переноса (CY) смещаются на один бит влево. Содержание CIO 010015 перемещается во Флаг переноса (CY), а содержание флага переноса смещается в CIO 010000.

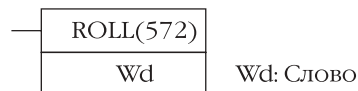




## 3-9-10 Команда вращения двойных данных влево DOUBLE ROTATE LEFT: ROLL(572)

**Назначение**

По команде ROLL(572) осуществляется смещение всех битов слов Wd и Wd+1, включая флаг переноса (CY), на один бит влево.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	ROLL(572)
	Выполняется один раз для дифференцирования вверх	@ ROLL(572)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

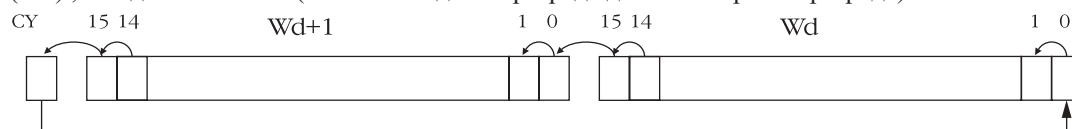
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142
Рабочая область	W000...W510
Область удержания бита (Holding bit area)	H000...H510
Область вспомогательных битов	A448...A958
Область таймера	T0000...T4094
Область счетчика	C0000...C4094
Область DM	D00000...D32766
Область EM, не содержащая банков	E00000...E32766
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)
Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-)IR15

**Описание**

По команде ROLL(572) осуществляется смещение всех битов слов Wd и Wd+1, включая флаг переноса (CY), на один бит влево (от бита младшего разряда до бита старшего разряда).

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF.

Наименование	Обозначение	Функционирование
Флаг равенства	=	Переводится в состояние ON, когда результатом сдвига является 0. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении смещения во Флаг переноса (CY) заносится 1. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда при выполнении смещения в старший бит заносится 1. OFF в других случаях.

### Меры предосторожности

После выполнения команды ROLL(572) флаг ошибки переводится в состояние OFF.

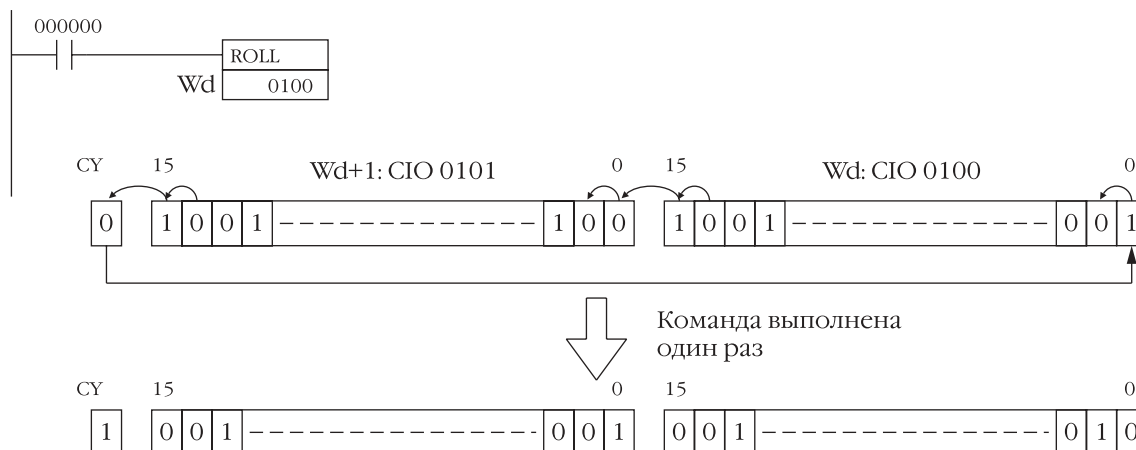
Если в результате смещения данных в словах Wd и Wd+1 содержатся нулевые значения, флаг равенства переводится в состояние ON.

Если в результате выполнения смещения данных содержанием бита старшего разряда слова Wd+1 становится 1, Флаг отрицательного значения переводится в состояние ON.

**Примечание:** Используя команды Set Carry (STC(040)) и Clear Carry (CLC(041)) можно установить флаг переноса в состояние 1 или 0 непосредственно перед выполнением команды ROL(027).

### Примеры

Когда бит CIO 000000 переводится в состояние ON, слова CIO 0100, CIO 0101 и флаг переноса смещаются на один бит влево. Содержание CIO 010015 перемещается во Флаг переноса (CY), а содержание флага переноса смещается в CIO 010000.

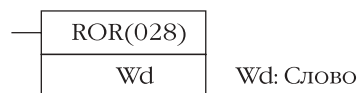


### 3-9-11 Команда вращения вправо ROTATE RIGHT: ROR(028)

#### Назначение

По команде ROR(028) осуществляется смещение всех битов слова Wd, включая флаг переноса (CY), на один бит вправо.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	ROR(028)
	Выполняется один раз для дифференцирования вверх	@ ROR(028)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

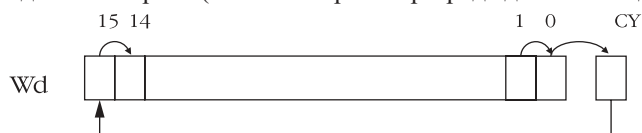
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Wd
Область ввода/вывода (область СЮ)	СЮ 0000...СЮ 6143
Рабочая область	W000...W511
Область удержания бита (Holding bit area)	H000...H511
Область вспомогательных битов	A448...A959
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область DM	D00000...D32767
Область EM, не содержащая банков	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)
Константы	—
Регистры данных	DR0...DR15
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,(- -)IR0...(- -)IR15

**Описание**

По команде ROR(028) осуществляется смещение всех битов слова Wd, включая флаг переноса (CY), на один бит вправо (от бита старшего разряда до бита младшего разряда).

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF.
Флаг равенства	=	Переводится в состояние ON, когда результатом сдвига является 0. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении смещения во Флаг переноса (CY) заносится 1. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда при выполнении смещения в старший бит заносится 1. OFF в других случаях.

**Меры предосторожности**

После выполнения команды ROR(028) флаг ошибки переводится в состояние OFF.

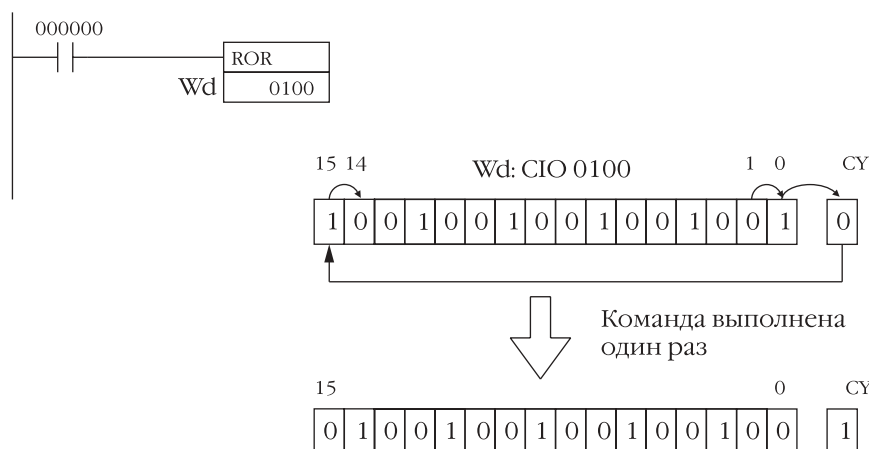
Если в результате смещения данных в слове Wd содержится нулевое значение, флаг равенства переводится в состояние ON.

Если в результате выполнения смещения данных содержанием бита старшего разряда слова Wd становится 1, Флаг отрицательного значения переводится в состояние ON.

**Примечание:** Используя команды Set Carry (STC(040)) и Clear Carry (CLC(041)) можно установить флаг переноса в состояние 1 или 0 непосредственно перед выполнением команды ROR(028).

**Примеры**

Когда бит CIO 000000 переводится в состояние ON, слово CIO 0100 и флаг переноса смещаются на один бит вправо. Содержание CIO 010000 перемещается во Флаг переноса (CY), а содержание флага переноса смещается в CIO 010015.

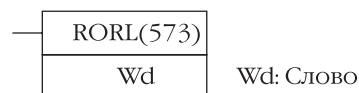


3-9-12 Команда вращения двойных данных вправо DOUBLE ROTATE RIGHT: RORL(573)

**Назначение**

По команде RORL(573) осуществляется смещение всех битов слов Wd и Wd+1, включая флаг переноса (CY), на один бит вправо.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	RORL(573)
	Выполняется один раз для дифференцирования вверх	@ RORL(573)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается.	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

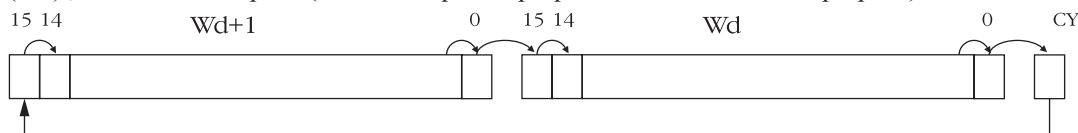
**Спецификации операндов**

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142
Рабочая область	W000...W510
Область удержания бита (Holding bit area)	H000...H510
Область вспомогательных битов	A448...A958
Область таймера	T0000...T4094
Область счетчика	C0000...C4094
Область DM	D00000...D32766
Область EM, не содержащая банков	E00000...E32766
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)
Константы	-

Область	Wd
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...IR15(++) ,(- -)IR0...(- -)IR15

### Описание

По команде RORL(573) осуществляется смещение всех битов слов Wd и Wd+1, включая флаг переноса (CY), на один бит вправо (от бита старшего разряда до бита младшего разряда).



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF.
Флаг равенства	=	Переводится в состояние ON, когда результатом сдвига является 0. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении смещения во Флаг переноса (CY) заносится 1. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда при выполнении смещения в старший бит заносится 1. OFF в других случаях.

### Меры предосторожности

После выполнения команды RORL(573) флаг ошибки переводится в состояние OFF.

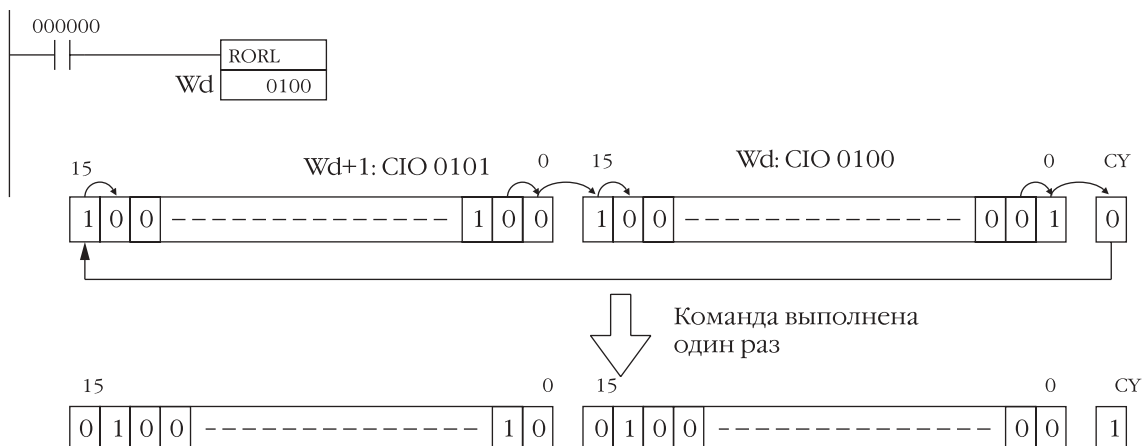
Если в результате смещения данных в словах Wd и Wd+1 содержатся нулевые значения, флаг равенства переводится в состояние ON.

Если в результате выполнения смещения данных содержанием бита старшего разряда слова Wd+1 становится 1, Флаг отрицательного значения переводится в состояние ON.

**Примечание:** Используя команды Set Carry (STC(040)) и Clear Carry (CLC(041)) можно установить флаг переноса в состояние 1 или 0 непосредственно перед выполнением команды RORL(573).

### Примеры

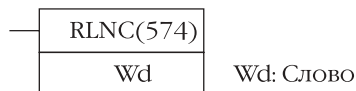
Когда бит CIO 000000 переводится в состояние ON, слова CIO 0100, CIO 0101 и флаг переноса смещаются на один бит вправо. Содержание CIO 010000 перемещается во Флаг переноса (CY), а содержание флага переноса смещается в CIO 010015.



## 3-9-13 Команда вращения влево без переноса ROTATE LEFT WITHOUT CARRY: RLNC(574)

**Назначение**

По команде RLNC(574) осуществляется смещение всех битов слова Wd, исключая флаг переноса (CY), на один бит влево.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	RLNC(574)
	Выполняется один раз для дифференцирования вверх	@ RLNC(574)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

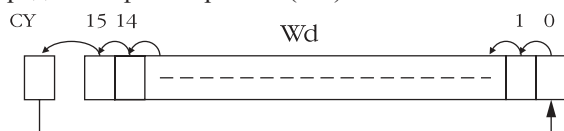
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143
Рабочая область	W000...W511
Область удержания бита (Holding bit area)	H000...H511
Область вспомогательных битов	A448...A959
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область DM	D00000...D32767
Область EM, не содержащая банков	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)
Константы	—
Регистры данных	DR0...DR15
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-(-)IR15

**Описание**

По команде RLNC(574) осуществляется смещение всех битов слова Wd (от бита младшего разряда до бита старшего разряда) влево. Содержание бита старшего разряда слова Wd перемещается в бит младшего разряда и во флаг переноса (CY).

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF.

Наименование	Обозначение	Функционирование
Флаг равенства	=	Переводится в состояние ON, когда результатом сдвига является 0. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении смещения во Флаг переноса (CY) заносится 1. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда при выполнении смещения в старший бит заносится 1. OFF в других случаях.

### Меры предосторожности

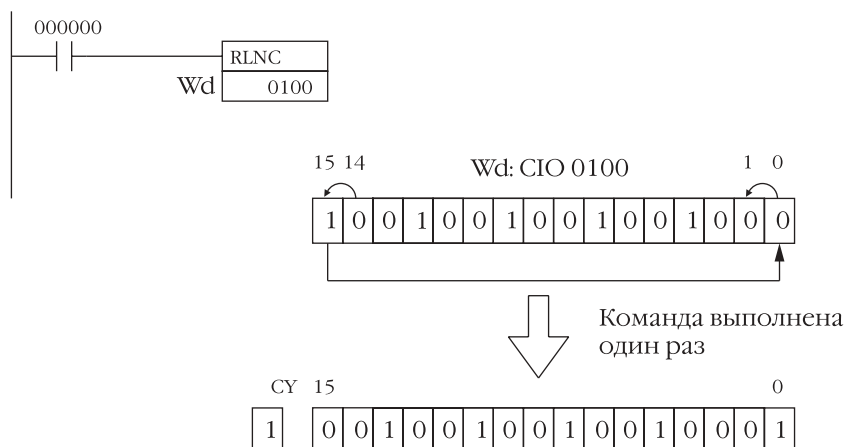
После выполнения команды RLNC(574) флаг ошибки переводится в состояние OFF.

Если в результате смещения данных в слове Wd содержится нулевое значение, флаг равенства переводится в состояние ON.

Если в результате выполнения смещения данных содержанием бита старшего разряда слова Wd становится 1, Флаг отрицательного значения переводится в состояние ON.

### Примеры

Когда бит CIO 000000 переводится в состояние ON, слово CIO 0100 смещается на один бит влево (исключая флаг переноса (CY)). Содержание CIO 010015 перемещается во Флаг переноса (CY), а содержание флага переноса смещается в CIO 010000.

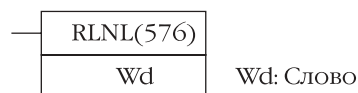


### 3-9-14 Команда вращения двойных данных влево без переноса DOUBLE ROTATE LEFT WITHOUT CARRY: RLNL(576)

#### Назначение

По команде RLNL(576) осуществляется смещение всех битов слов Wd и Wd+1, исключая флаг переноса (CY), на один бит влево.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	RLNL(576)
	Выполняется один раз для дифференцирования вверх.	@ RLNL(576)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

#### Применяемые области программы

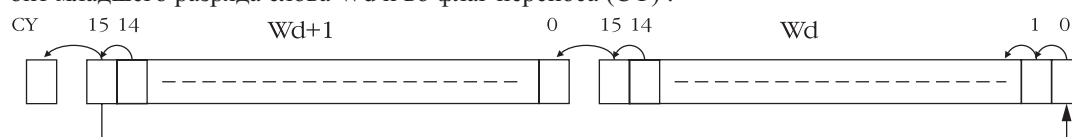
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

### Спецификации операндов

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142
Рабочая область	W000...W510
Область удержания бита (Holding bit area)	H000...H510
Область вспомогательных битов	A448...A958
Область таймера	T0000...T4094
Область счетчика	C0000...C4094
Область DM	D00000...D32766
Область EM, не содержащая банков	E00000...E32766
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)
Константы	-
Регистры данных	-
Индексные регистры	-
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-)IR15

### Описание

По команде RLNL(576) осуществляется смещение всех битов слов Wd и Wd+1 на один бит влево (от бита младшего разряда до бита старшего разряда). Содержание бита старшего разряда слова Wd+1 смещается в бит младшего разряда слова Wd и во флаг переноса (CY).



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF.
Флаг равенства	=	Переводится в состояние ON, когда результатом сдвига является 0. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении смещения во Флаг переноса (CY) заносится 1. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда при выполнении смещения в старший бит заносится 1. OFF в других случаях.

### Меры предосторожности

После выполнения команды RLNL(576) флаг ошибки переводится в состояние OFF.

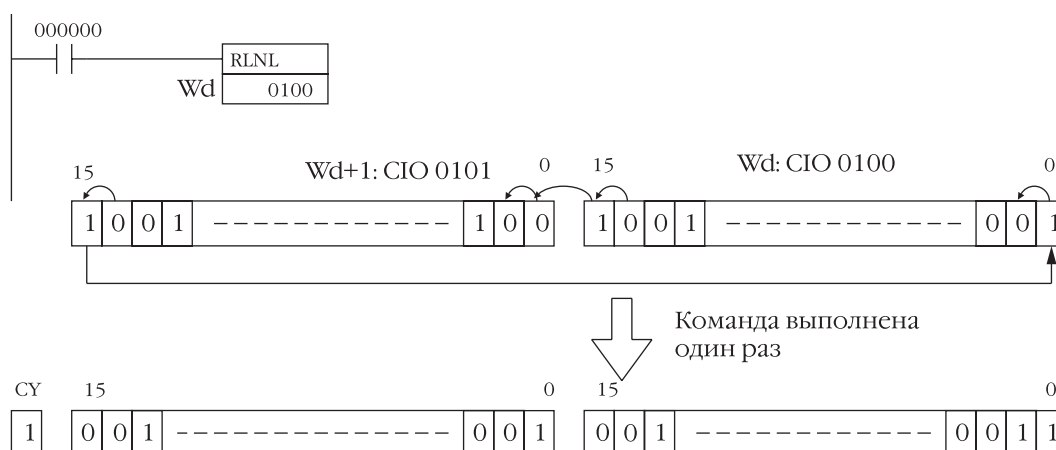
Если в результате смещения данных в словах Wd и Wd+1 содержатся нулевые значения, флаг равенства переводится в состояние ON.

Если в результате выполнения смещения данных содержанием бита старшего разряда слова Wd+1 становится 1, Флаг отрицательного значения переводится в состояние ON.

### Примеры

Когда бит CIO 000000 переводится в состояние ON, слова CIO 0100, CIO 0101 (исключая флаг переноса (CY)) смещаются на один бит влево. Содержание CIO 010015 перемещается в CIO 010000.



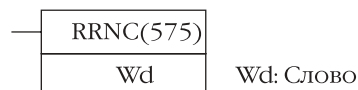


### 3-9-15 Команда вращения вправо без переноса ROTATE RIGHT WITHOUT CARRY: RRNC(575)

#### Назначение

По команде RRNC(575) осуществляется смещение всех битов слова Wd, исключая флаг переноса (CY), на один бит вправо. Содержание бита младшего разряда слова Wd смещается в бит старшего разряда и во флаг переноса.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	RRNC(575)
	Выполняется один раз для дифференцирования вверх	@ RRNC(575)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

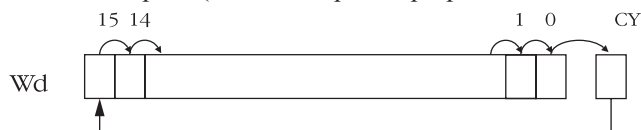
#### Спецификации операндов

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143
Рабочая область	W000...W511
Область удержания бита (Holding bit area)	H000...H511
Область вспомогательных битов	A448...A959
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область DM	D00000...D32767
Область EM, не содержащая банков	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)
Константы	—
Регистры данных	DR0...DR15
Индексные регистры	—

Область	Wd
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15

**Описание**

По команде RRNC(575) осуществляется смещение всех битов слова Wd, исключая флаг переноса (CY), на один бит вправо (от бита старшего разряда до бита младшего разряда).



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF.
Флаг равенства	=	Переводится в состояние ON, когда результатом сдвига является 0. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении смещения во Флаг переноса (CY) заносится 1. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда при выполнении смещения в старший бит заносится 1. OFF в других случаях.

**Меры предосторожности**

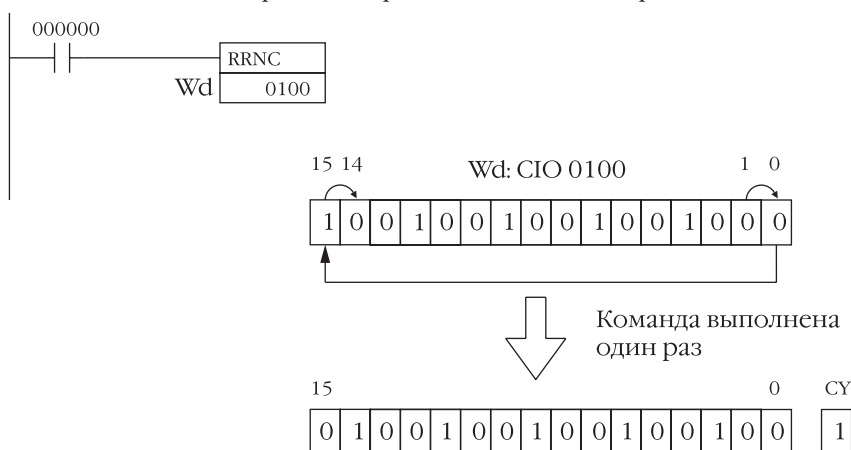
После выполнения команды RRNC(575) флаг ошибки переводится в состояние OFF.

Если в результате смещения данных в слове Wd содержится нулевое значение, флаг равенства переводится в состояние ON.

Если в результате выполнения смещения данных содержанием бита старшего разряда слова Wd становится 1, Флаг отрицательного значения переводится в состояние ON.

**Примеры**

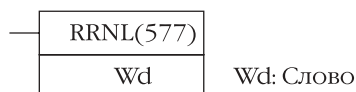
Когда бит CIO 000000 переводится в состояние ON, слово CIO 0100, исключая флаг переноса (CY), смещается на один бит вправо. Содержание CIO 010000 перемещается в CIO 010015.



**3-9-16 Команда вращения двойных данных вправо без переноса DOUBLE ROTATE RIGHT WITHOUT CARRY: RRNL(577)**

**Назначение**

По команде RRNL(577) осуществляется смещение всех битов слов Wd и Wd+1, исключая флаг переноса (CY), на один бит вправо. Содержание бита младшего разряда слова Wd+1 смещается в бит старшего разряда слова Wd и во флаг переноса.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	RRNL(577)
	Выполняется один раз для дифференцирования вверх	@ RRNL(577)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

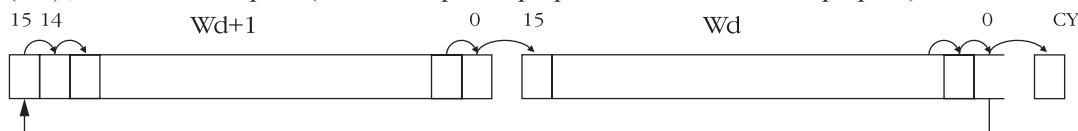
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142
Рабочая область	W000...W510
Область удержания бита (Holding bit area)	H000...H510
Область вспомогательных битов	A448...A958
Область таймера	T0000...T4094
Область счетчика	C0000...C4094
Область DM	D00000...D32766
Область EM, не содержащая банков	E00000...E32766
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)
Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15(++ ,(- -)IR0...,(- -)IR15

**Описание**

По команде RRNL(577) осуществляется смещение всех битов слов Wd и Wd+1, исключая флаг переноса (CY), на один бит вправо (от бита старшего разряда до бита младшего разряда).

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF.
Флаг равенства	=	Переводится в состояние ON, когда результатом сдвига является 0. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении смещения во Флаг переноса (CY) заносится 1. OFF в других случаях.

Наименование	Обозначение	Функционирование
Флаг отрицательного значения	N	Переводится в состояние ON, когда при выполнении смещения в старший бит заносится 1. OFF в других случаях.

### Меры предосторожности

После выполнения команды RRNL(577) флаг ошибки переводится в состояние OFF.

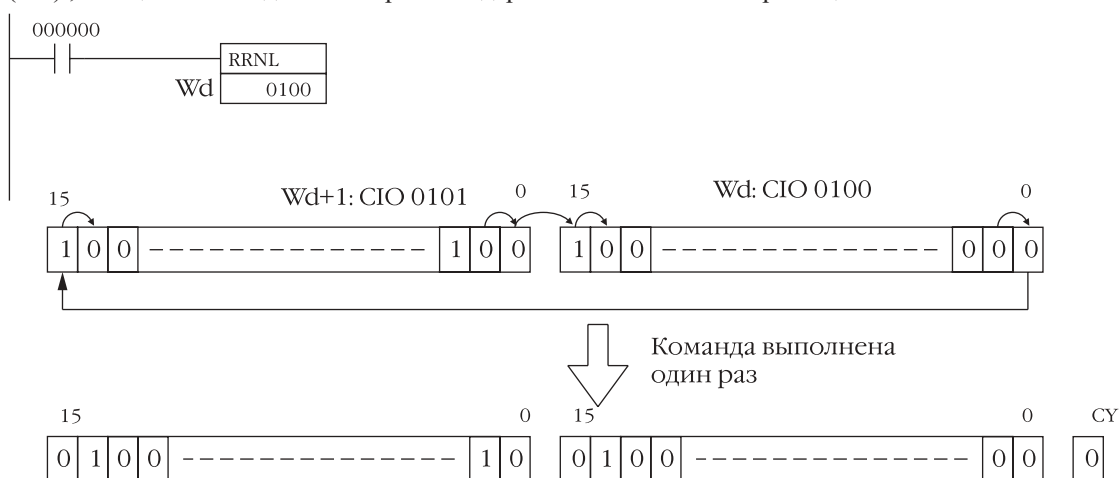
Если в результате смещения данных в словах Wd и Wd+1 содержатся нулевые значения, флаг равенства переводится в состояние ON.

Если в результате выполнения смещения данных содержанием бита старшего разряда слова Wd+1 становится 1, Флаг отрицательного значения переводится в состояние ON.

**Примечание:** Используя команды Set Carry (STC(040)) и Clear Carry (CLC(041)) можно установить флаг переноса в состояние 1 или 0 непосредственно перед выполнением команды RRNL(577).

### Примеры

Когда бит CIO 000000 переводится в состояние ON, слова CIO 0100, CIO 0101, исключая флаг переноса (CY), смещаются на один бит вправо. Содержание CIO 010000 перемещается в CIO 010015.



### 3-9-17 Команда перемещения на одну цифру влево ONE DIGIT SHIFT LEFT: SLD(074)

#### Назначение

По команде SLD(074) осуществляется смещение данных на одну цифру влево (4 бита).

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	SLD(074)
	Выполняется один раз для дифференцирования вверх	@ SLD(074)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

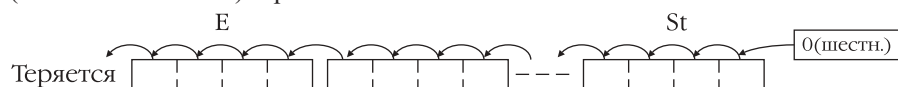
#### Спецификации операндов

Область	St	E
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	

Область	St	E
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A448...A959	
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	—	
Регистры данных	—	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048+2047,IR0-2048...+2047,IR15 DR0...DR15,IR0...IR15 ,IR0+(++)...,IR15+(++) ,(-)IR0...,(-)IR15	

### Описание

По команде SLD(074) осуществляется смещение всех данных между St и E на одну цифру (4 бита) влево. В цифру младшего разряда (биты 3...0 слова St) помещаются нули, а содержание цифры старшего разряда (биты 15 12 слова E) теряется.



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда St больше E. OFF в других случаях

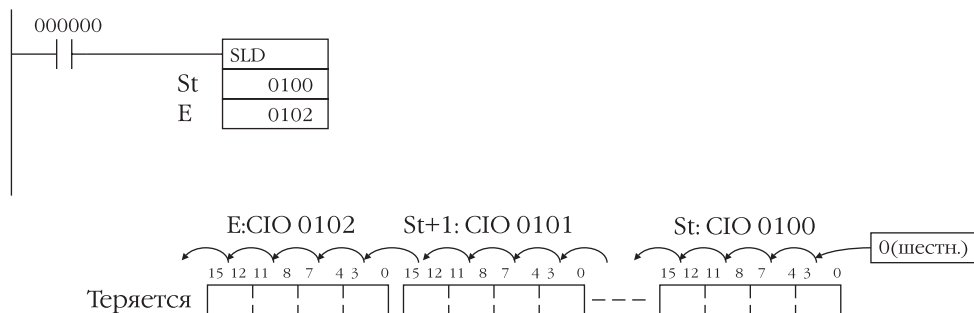
### Меры предосторожности

В случае, когда St больше E, определяется ошибка, и флаг ошибки переводится в состояние ON.

**Примечание:** При смещении большого количества данных время выполнения команды достаточно велико. Предпринимайте меры по предотвращению отключения питания при выполнении команды SLD(074) во избежание прерывания выполнения операции смещения данных до ее полного завершения.

### Примеры

Когда бит CIO 000000 переводится в состояние ON, слова от CIO 0100 до CIO 0102 смещаются на одну цифру (4 бита) влево. В биты 3...0 слова CIO 0100 помещаются нули, а содержание битов 15 12 слова CIO 0102 теряется.



## 3-9-18 Команда перемещения на одну цифру вправо ONE DIGIT SHIFT RIGHT: SRD(075)

**Назначение**

По команде SRD(075) осуществляется смещение данных на одну цифру вправо (4 бита).

**Символ релейно-контактной схемы**

SRD(075)	
St	St: Начальное слово
E	E: Завершающее слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	SRD(075)
	Выполняется один раз для дифференцирования вверх	@ SRD(075)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Примечание:** Слова St и E должны находиться в одной области данных.

**Спецификации операндов**

Область	St	E
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A448...A959	
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	-	
Регистры данных	-	
Индексные регистры	-	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15	

**Описание**

По команде SRD(075) осуществляется смещение всех данных между St и E на одну цифру (4 бита) вправо. В цифру старшего разряда (биты 15...12 слова E) помещаются нули, а содержание цифры младшего разряда (биты 3...0 слова St) теряется.



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда St больше E. OFF в других случаях

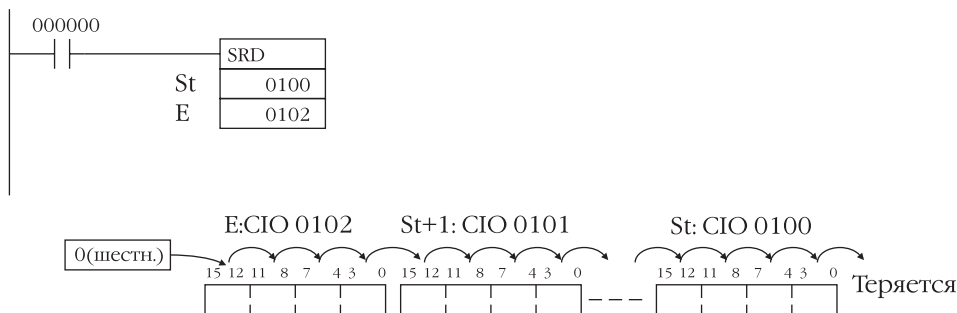
**Меры предосторожности**

В случае, когда St больше E, определяется ошибка, и флаг ошибки переводится в состояние ON. После выполнения команды SRD(075) флаги равенства и флаг отрицательного значения переводятся в состояние OFF.

***Примечание:** При смещении большого количества данных время выполнения команды достаточно велико. Предпринимайте меры по предотвращению отключения питания при выполнении команды SRD(075) во избежание прерывания выполнения операции смещения данных до ее полного завершения.*

**Примеры**

Когда бит CIO 000000 переводится в состояние ON, слова от CIO 0100 до CIO 0102 смещаются на одну цифру (4 бита) вправо. В биты 15...12 слова CIO 0102 помещаются нули, а содержание битов 0...3 слова CIO 0100 теряется.



3-9-19 Команда смещения N битов данных влево SHIFT N-BIT DATA LEFT: NSFL(578)

**Назначение**

По команде NSFL(578) осуществляется смещение заданного количества битов влево.

**Символ релейно-контактной схемы**

NSFL(578)	
D	D: Начальное слово при выполнении смещения
C	C: Начальный бит
N	N: Количество битов

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	NSFL(578)
	Выполняется один раз для дифференцирования вверх	@ NSFL(578)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды**

C: 0000...000F (шестн.) (0...15)  
N: 0000...FFFF (шестн.) (0...65535)

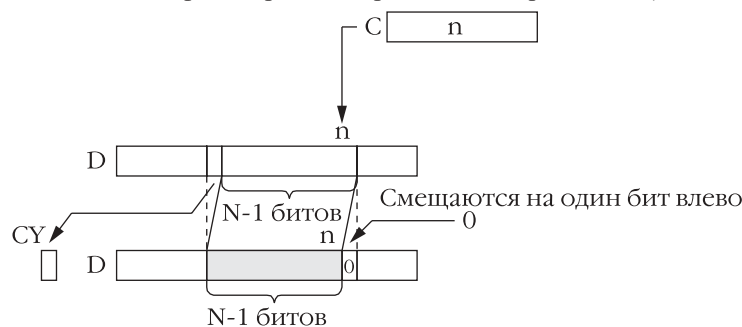
***Примечание:** Слова в перемещаемом регистре должны находиться в одной области данных.*

### Спецификации операндов

Область	D	C	N
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A448...A959	A000...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	–	#0000...#000F (двоичные) или &0...&15	#0000...#FFFF (двоичные) или &0...&65535
Регистры данных	–	DR0...DR15	
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-(-)IR15		

### Описание

По команде NSFL(578) осуществляется смещение заданного количества битов N в слове D, начиная от бита C, на один бит влево (по направлению к слову старшего разряда и биту старшего разряда). В начальный бит C заносится «0», а содержание старшего бита из указанного количества битов перемещается во флаг переноса (CY). (Таким образом, фактически на один бит перемещается N-1 битов, а старший бит записывается во флаг переноса. Примечание переводчика.)



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда данные C находятся за пределами диапазона от 0000 до 000F (шестн.). OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда 1 заносится во флаг переноса (CY). OFF в других случаях.

### Меры предосторожности

В случае, когда длина перемещаемых данных равна нулю, содержание начального бита записывается во флаг переноса (CY), при этом содержание начального бита не изменяется.

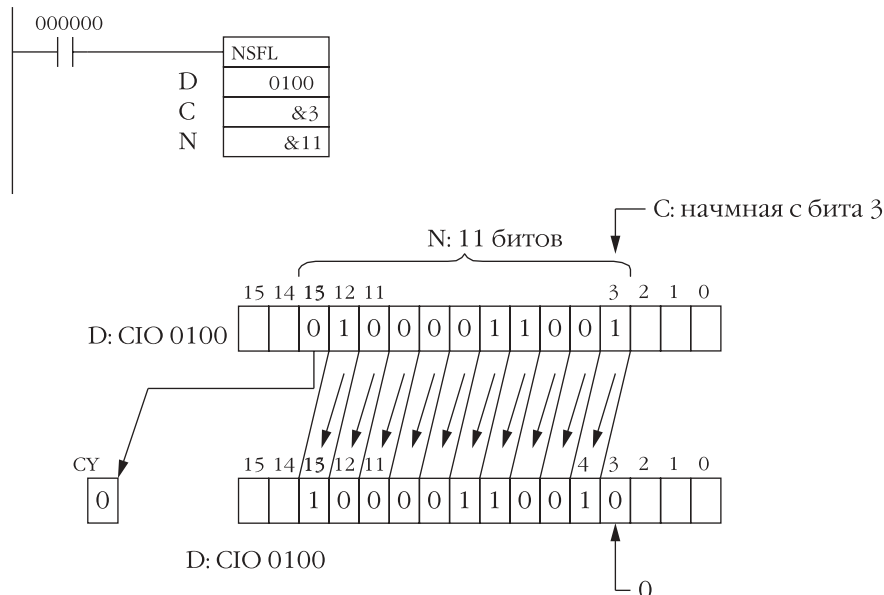
Таким образом, подвергаются изменению только биты, перемещаемые влево в указанном слове младшего разряда (т.е. изменяются только данные указанного слова)



**Примеры**

Когда бит СЮ 000000 переводится в состояние ON, все биты, начиная от бита 3, до бита, определяемого длиной перемещаемых данных (В шестн.) (& 11 прим. переводчика) смещаются на один бит влево (от бита младшего разряда до бита старшего разряда).

В бит 3 слова СЮ 0100 заносится «0». Содержание бита старшего разряда в перемещаемой области (бит 13 слова СЮ 0100) записывается во флаг переноса (СУ).

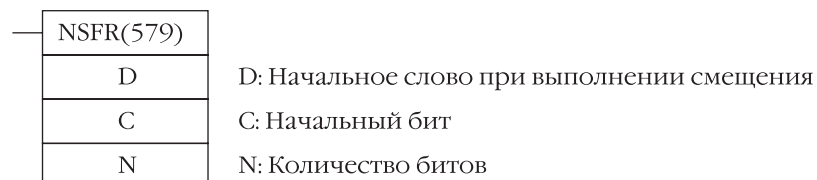


3-9-20 Команда смещения N битов данных вправо SHIFT N-BIT DATA RIGHT: NSFR(579)

**Назначение**

По команде NSFR(579) осуществляется смещение заданного количества битов вправо.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	NSFR(579)
	Выполняется один раз для дифференцирования вверх	@ NSFR(579)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды**

**С:** 0000...000F (шестн.) (0...15)

**N:** 0000...FFFF (шестн.) (0...65535)

*Примечание:* Слова в перемещаемом регистре должны находиться в одной области данных.

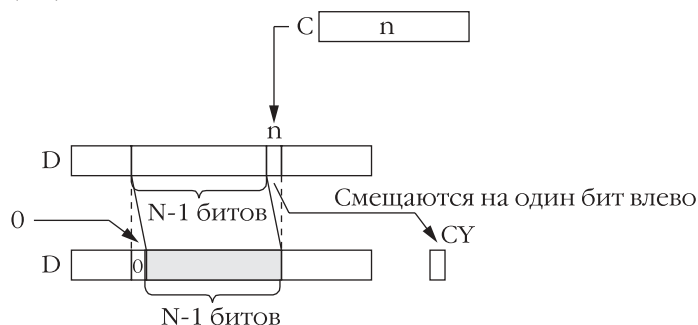
**Спецификации операндов**

Область	D	C	N
Область ввода/вывода (область СЮ)	СЮ 0000...СЮ 6143		

Область	D	C	N
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A448...A959	A000...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	—	#0000...#000F (двоичные) или &0...&15	#0000...#FFFF (двоичные) или &0...&65535
Регистры данных	—	DR0...DR15	
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-(-)IR15		

### Описание

По команде NSFR(579) осуществляется смещение заданного количества битов N, начиная от бита C в слове D, на один бит вправо (по направлению к слову младшего разряда и биту младшего разряда). В начальный бит заносится «0», а содержание младшего бита в смещаемой области перемещается во флаг переноса (CY).



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда данные C находятся за пределами диапазона от 0000 до 000F (шестн.). OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда 1 заносится во флаг переноса (CY). OFF в других случаях.

### Меры предосторожности

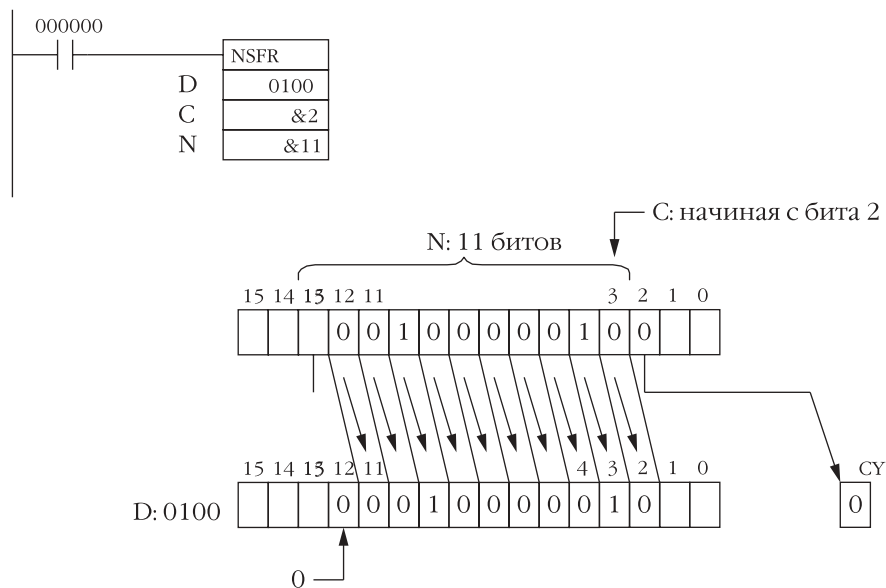
В случае, когда длина перемещаемых данных равна нулю, содержание начального бита записывается во флаг переноса (CY), при этом содержание начального бита не изменяется.

Таким образом, подвергаются изменению только биты, перемещаемые вправо в указанном слове (т.е. изменяются только данные указанного слова).

### Примеры

Когда бит CIO 000000 переводится в состояние ON, все биты, начиная от бита 2, до конца перемещаемых данных (длиной 11 битов (В шестн.)) смещаются на один бит вправо (от бита старшего разряда до бита младшего разряда).

В бит 12 слова CIO 0100 заносится «0». Содержание бита младшего разряда в перемещаемой области (бит 2 слова CIO 0100) записывается во флаг переноса (CY).

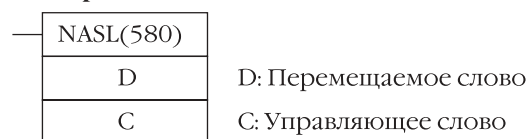


### 3-9-21 Команда смещения данных на N битов влево SHIFT N-BIT LEFT: NASL(580)

#### Назначение

По команде NASL(580) осуществляется смещение 16-ти битов указанного слова влево на заданное количество битов.

#### Символ релейно-контактной схемы

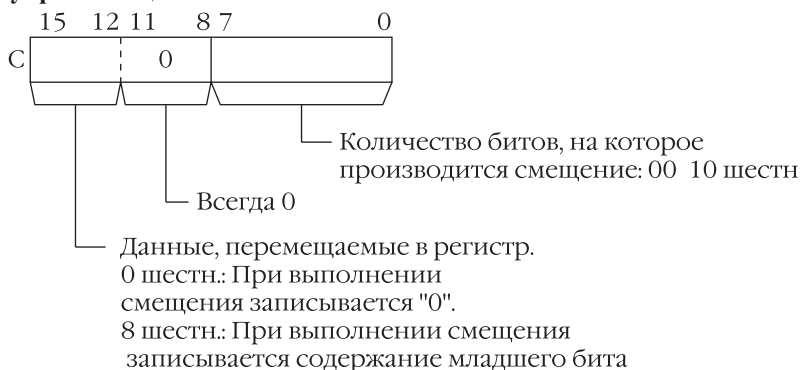


#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON.	NASL(580)
	Выполняется один раз для дифференцирования вверх	@ NASL(580)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

#### Применяемые области программы

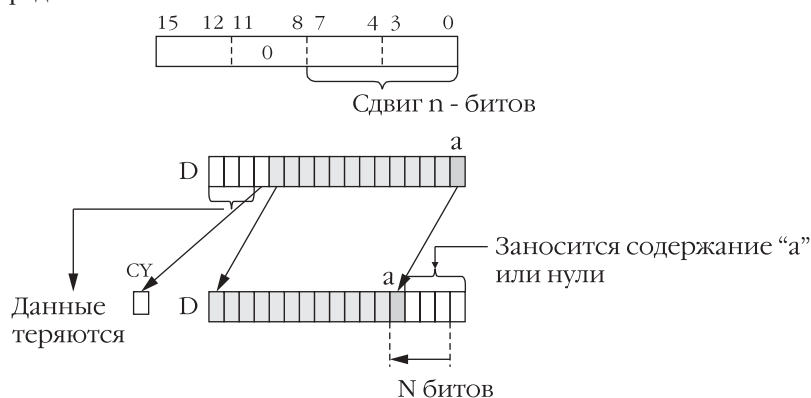
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****C: управляющее слово.****Спецификации операндов**

Область	D	C
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A448...A959	A000...A959
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	–	Только заданные значения
Регистры данных	DR0...DR15	
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15(++)) ,-(-)IR0...,(-)IR15	

**Описание**

По команде NASL(580) производится сдвиг перемещаемого слова D влево на заданное количество двойных битов (указанных в C) (от бита младшего разряда к биту старшего разряда). Либо нули, либо значение бита младшего разряда заносится вместо битов перемещаемого слова, начиная с бита младшего разряда.



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда управляющее слово С (количество битов смещения) находится за пределами допустимых пределов. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда результатом сдвига является 0. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении смещения во Флаг переноса (CY) заносится 1. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда при выполнении смещения в старший бит записывается 1. OFF в других случаях.

**Меры предосторожности**

Все биты, которые смещаются за пределы указанного слова, теряются, за исключением последнего из утрачиваемых битов, который записывается во флаг переноса (CY).

Когда количество битов смещения (указанное в С), равно нулю, смещение данных не производится. Тем не менее, соответствующие флаги переводятся в состояние ON или OFF, в зависимости от содержания указанных слов.

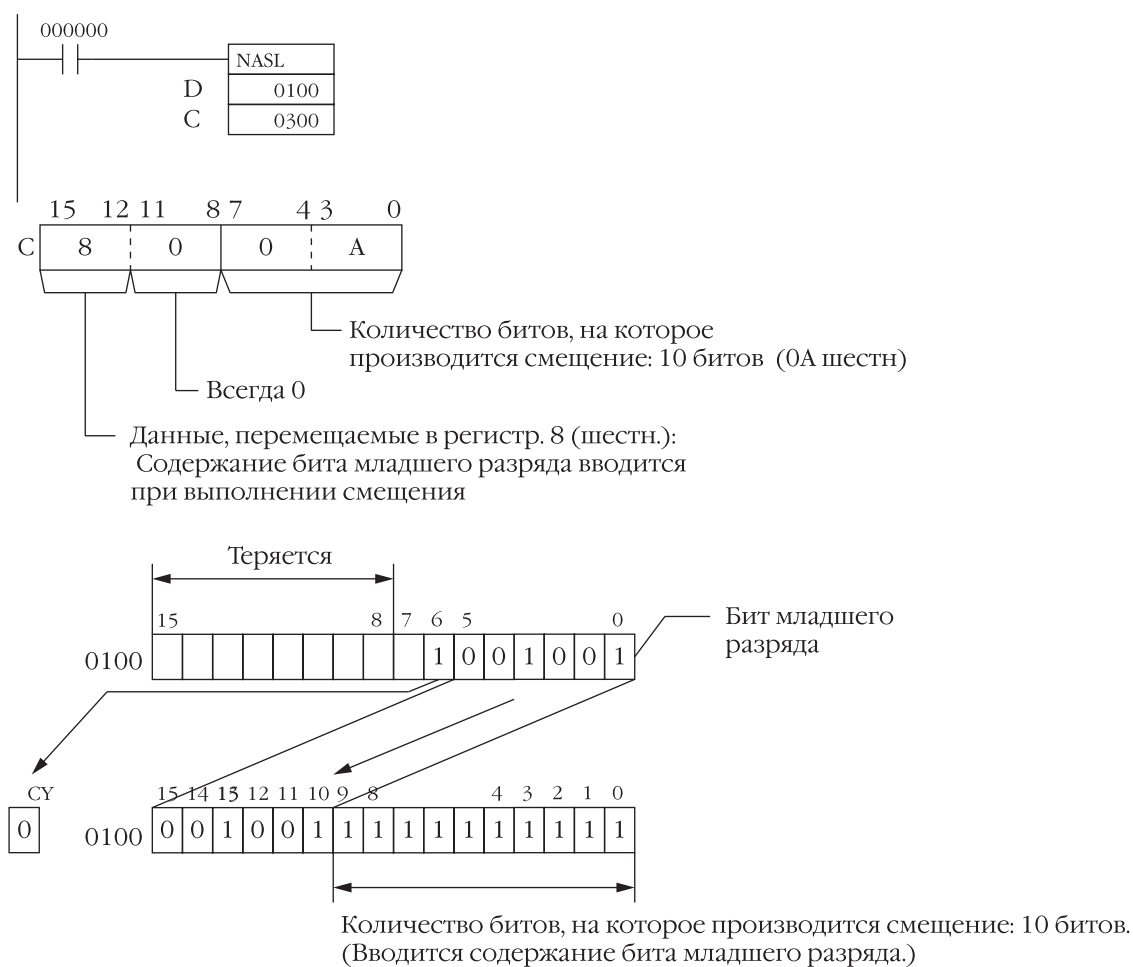
Если содержание управляющего слова С находится за пределами допускаемого диапазона, определяется ошибка и флаг ошибки переводится в состояние ON.

В случае, когда в результате выполнения смещения данных содержание D равно 0000 (шестн.), флаг равенства переводится в состояние ON.

Когда после выполнения смещения содержание бита старшего разряда слова D равно 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

Когда бит С10 000000 переводится в состояние ON, содержание С10 0100 смещается на 10 битов влево (от бита младшего разряда до бита старшего разряда). Количество битов, на которое производится смещение, указывается в битах 0...7 слова С10 0300. Содержание бита 0 слова С10 0100 записывается справа вместо смещенных битов, а содержание последнего из утрачиваемых битов, которые перемещаются за пределы указанного слова, записывается во флаг переноса (CY). Прочие данные теряются.

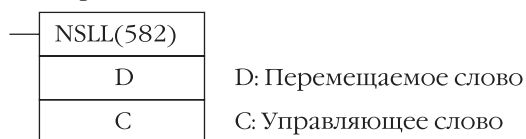


3-9-22 Команда смещения двойных данных на N битов влево DOUBLE SHIFT N-BIT LEFT: NSLL(582)

**Назначение**

По команде NSLL(582) осуществляется смещение 32-х битов указанных слов влево на заданное количество битов.

**Символ релейно-контактной схемы**

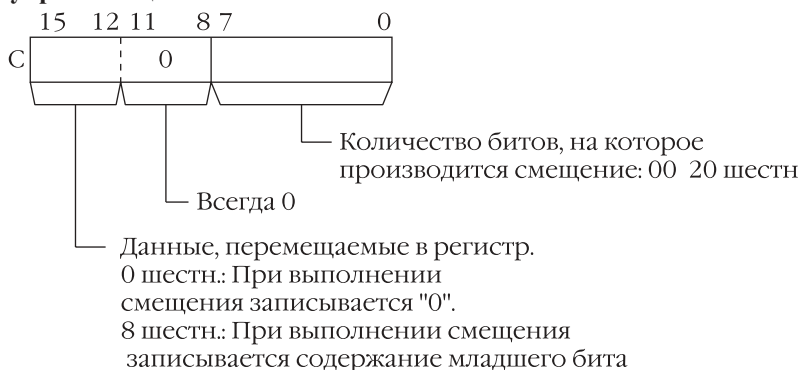


**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	NSLL(582)
	Выполняется один раз для дифференцирования вверх	@ NSLL(582)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

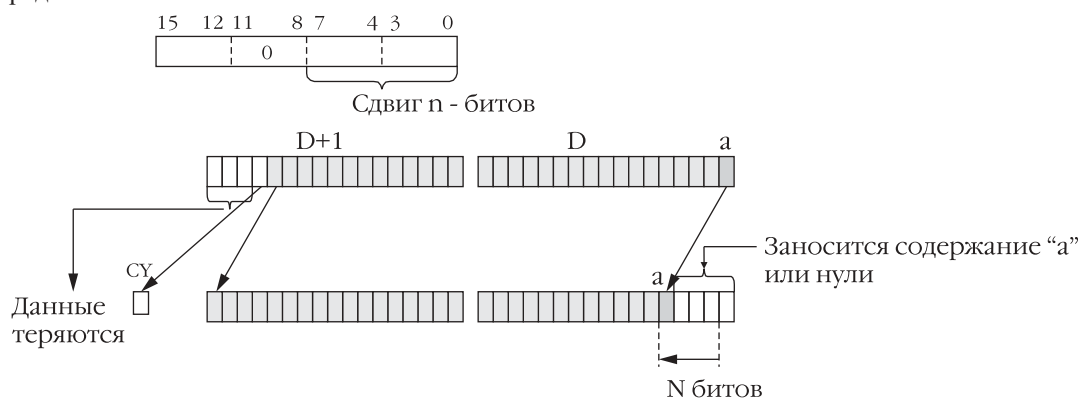
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****С: управляющее слово.****Спецификации операндов**

Область	D	C
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	CIO 0000...CIO 6143
Рабочая область	W000...W510	W000...W511
Область удержания бита (Holding bit area)	H000...H510	H000...H511
Область вспомогательных битов	A448...A958	A000...A959
Область таймера	T0000...T4094	T0000...T4095
Область счетчика	C0000...C4094	C0000...C4095
Область DM	D00000...D32766	D00000...D32767
Область EM, не содержащая банков	E00000...E32766	E00000...E32767
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	—	Только заданные значения
Регистры данных	—	DR0...DR15
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15	

**Описание**

По команде NSLL(582) осуществляется смещение перемещаемых слов D и D+1 влево на заданное в C количество битов (от бита младшего разряда к биту старшего разряда). Либо нули, либо значение младшего бита заносятся в указанное количество битов вместо перемещаемого слова, начиная с бита младшего разряда.



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда управляющее слово С (количество битов смещения) находится за пределами допустимых пределов. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда результатом сдвига является 0. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении смещения во Флаг переноса (CY) заносится 1. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда при выполнении смещения в старший бит записывается 1. OFF в других случаях.

**Меры предосторожности**

Все биты, которые смещаются за пределы указанного слова, теряются, за исключением последнего из утрачиваемых битов, который записывается во флаг переноса (CY)..

Когда количество битов смещения (указанное в С), равно нулю, смещение данных не производится. Тем не менее, соответствующие флаги переводятся в состояние ON или OFF, в зависимости от содержания указанных слов.

Если содержание управляющего слова С находится за пределами допускаемого диапазона, определяется ошибка и флаг ошибки переводится в состояние ON.

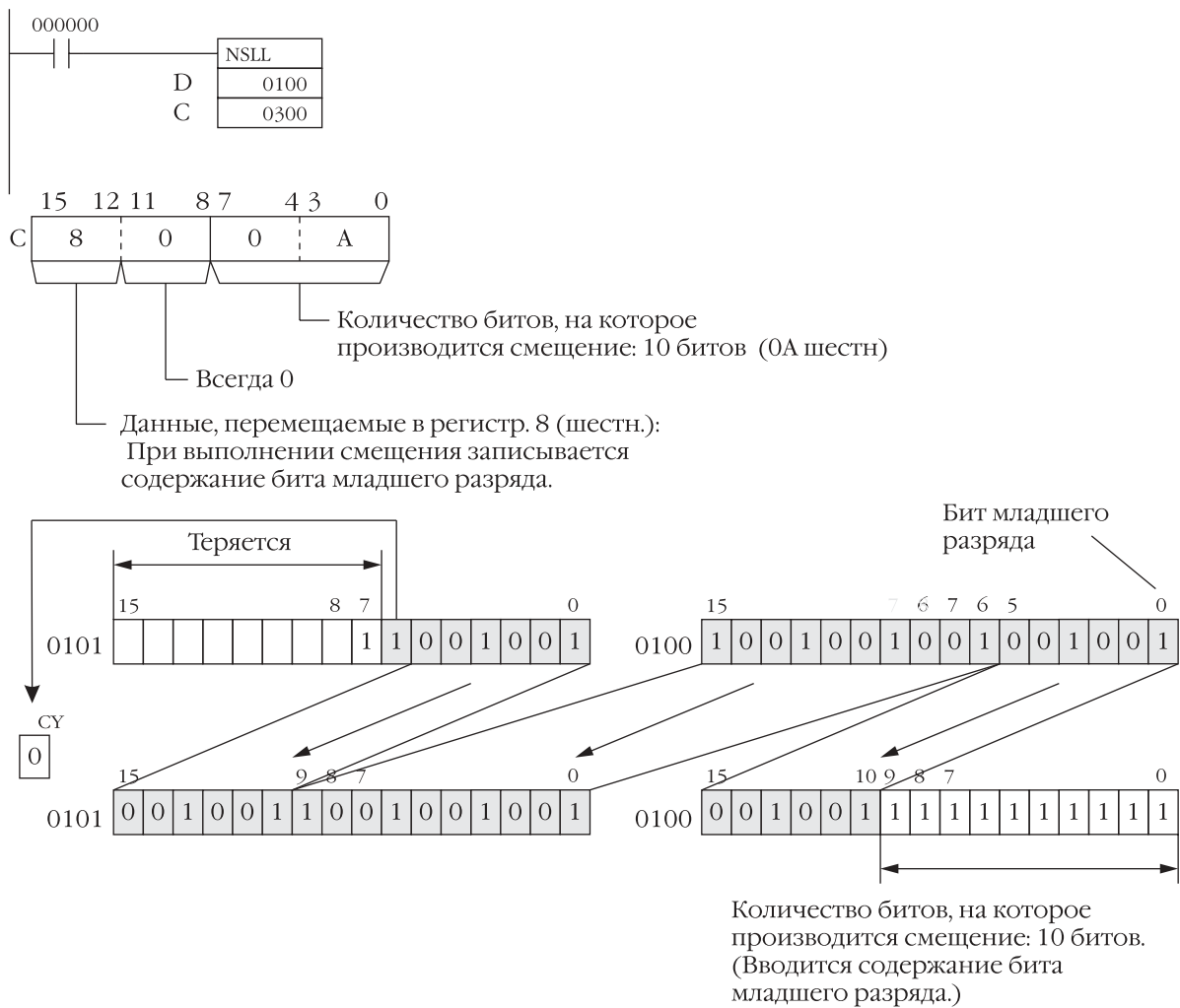
В случае, когда в результате выполнения смещения данных содержание D равно 0000, флаг равенства переводится в состояние ON.

В случае, когда в результате выполнения смещения данных содержание битов старшего разряда D и D+1 равно 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

Когда бит СЮ 000000 переводится в состояние ON, содержание СЮ 0100 и СЮ 0101 смещается на 10 битов влево (от бита младшего разряда к биту старшего разряда). Количество битов, на которое производится смещение, указывается в битах 0...7 слова СЮ 0300 (данные управления). Содержание бита 0 слова СЮ 0100 записывается справа вместо смещенных данных, а содержание последнего из теряемых битов, которые перемещаются за пределы указанного слова, записывается во флаг переноса (CY). Прочие данные теряются.



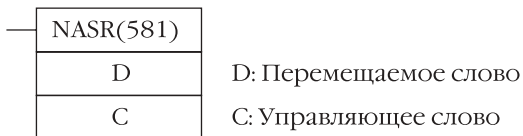


3-9-21 Команда сдвига данных на N битов вправо SHIFT N-BIT RIGHT: NASR(581)

**Назначение**

По команде NASR(581) осуществляется сдвиг 16-ти битов указанного слова вправо на заданное количество битов.

**Символ релейно-контактной схемы**

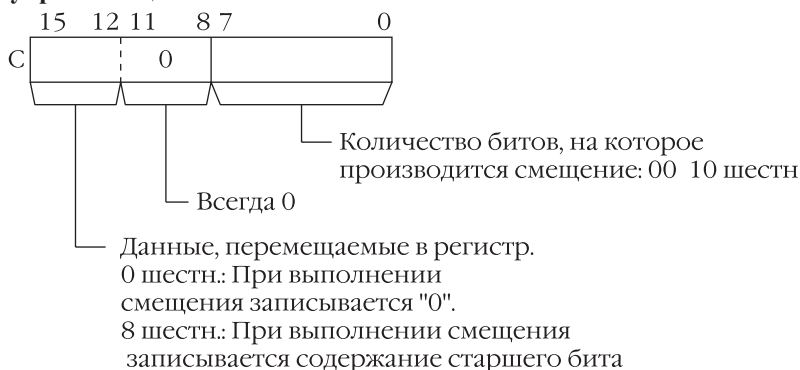


**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	NASR(581)
	Выполняется один раз для дифференцирования вверх	@ NASR(581)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

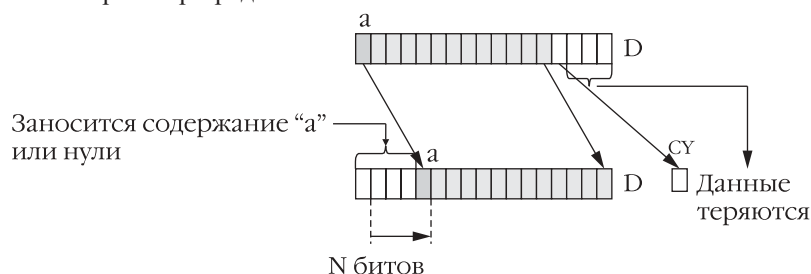
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****С: управляющее слово.****Спецификации операндов**

Область	D	C
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A448...A959	A000...A447, A448...A959
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	–	Только заданные значения
Регистры данных	DR0...DR15	
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15	

**Описание**

По команде NASR(581) производится сдвиг перемещаемого слова D вправо на заданное количество двойных битов (указанных в C) (от бита младшего разряда к биту старшего разряда). Либо нули, либо значение бита старшего разряда заносятся в указанное количество битов перемещаемого слова, начиная с бита старшего разряда.



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда управляющее слово С (количество битов смещения) находится за пределами допустимых пределов. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда результатом сдвига является 0. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении смещения во Флаг переноса (CY) заносится 1. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда при выполнении смещения в старший бит записывается 1. OFF в других случаях.

**Меры предосторожности**

Все биты, которые смещаются за пределы указанного слова, теряются, за исключением последнего из утрачиваемых битов, который записывается во флаг переноса (CY).

Когда количество битов смещения (указанное в С), равно нулю, смещение данных не производится. Тем не менее, соответствующие флаги переводятся в состояние ON или OFF, в зависимости от содержания указанных слов.

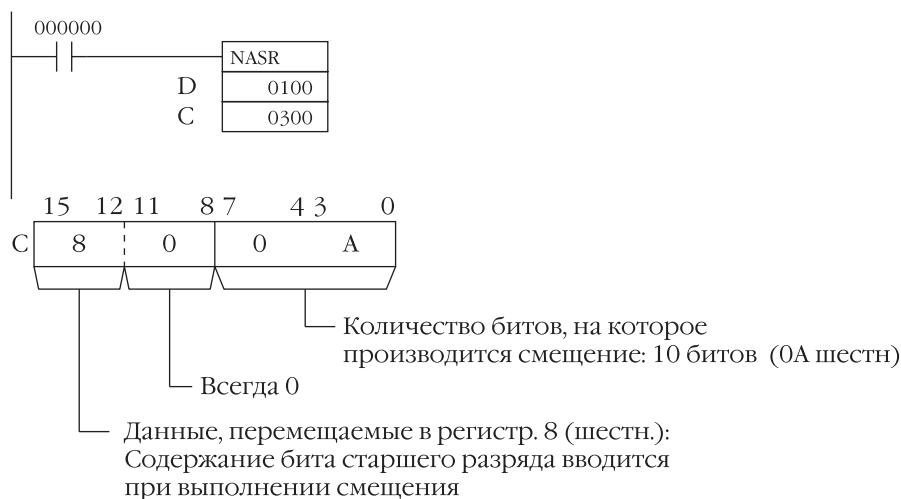
Если содержание управляющего слова С находится за пределами допускаемого диапазона, определяется ошибка и флаг ошибки переводится в состояние ON.

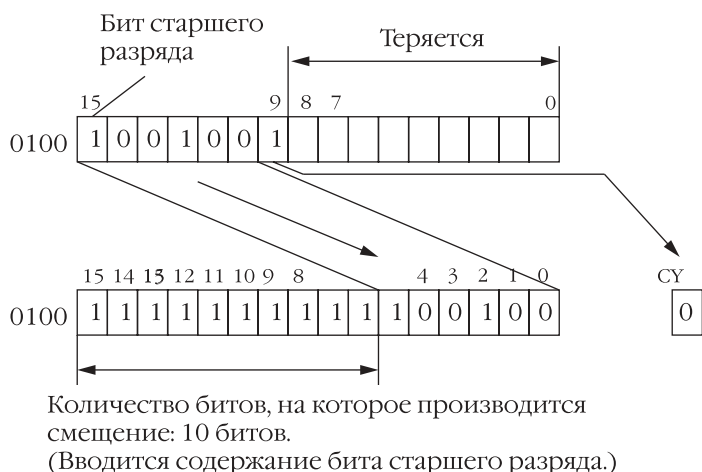
В случае, когда в результате выполнения смещения данных содержание D равно 0000 (шестн.), флаг равенства переводится в состояние ON.

Когда после выполнения смещения содержание бита старшего разряда слова D равно 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

Когда бит СЮ 000000 переводится в состояние ON, содержание СЮ 0100 смещается на 10 битов вправо (от бита старшего разряда до бита младшего разряда). Количество битов, на которое производится смещение, указывается в битах 0...7 слова СЮ 0300. Содержание бита 15 слова СЮ 0100 записывается слева вместо смещенных битов, а содержание последнего из утрачиваемых битов, которые перемещаются за пределы указанного слова, записывается во флаг переноса (CY). Прочие данные теряются.



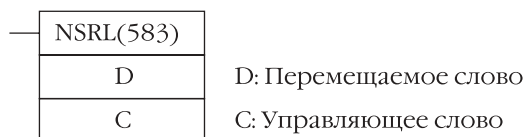


3-9-24 Команда смещения двойных данных на N битов вправо DOUBLE SHIFT N-BIT RIGHT: NSRL(583)

**Назначение**

По команде NSRL(583) осуществляется смещение 32-х битов указанных слов влево на заданное количество битов.

**Символ релейно-контактной схемы**



**Модификации**

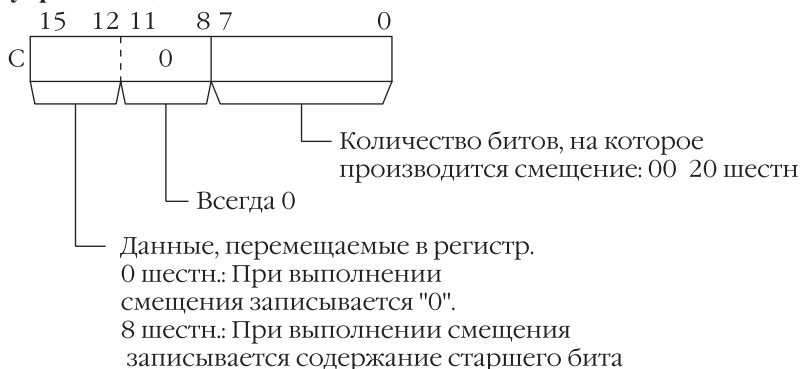
Модификации	Выполняется в каждом цикле для состояния ON.	NSRL(583)
	Выполняется один раз для дифференцирования вверх	@ NSRL(583)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды**

**C: управляющее слово.**



**Спецификации операндов**

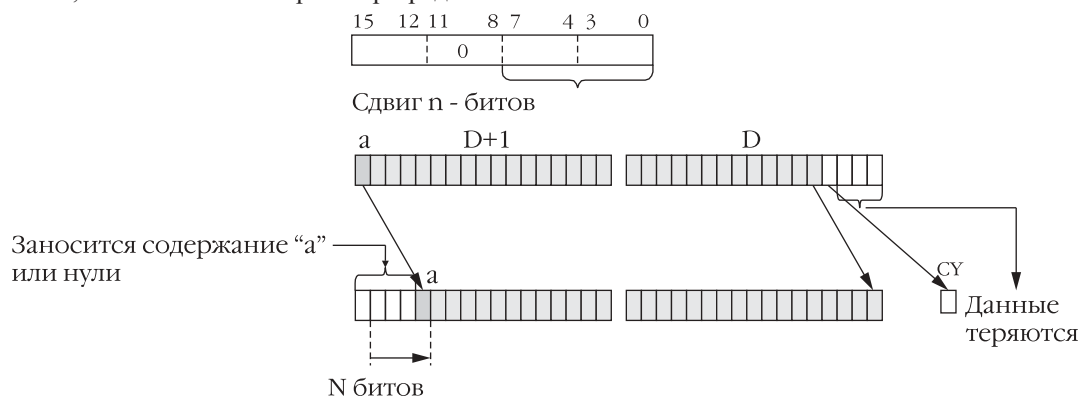
Область	D	C
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	CIO 0000...CIO 6143

Область	D	C
Рабочая область	W000...W510	W000...W511
Область удержания бита (Holding bit area)	H000...H510	H000...H511
Область вспомогательных битов	A448...A958	A000...A959
Область таймера	T0000...T4094	T0000...T4095
Область счетчика	C0000...C4094	C0000...C4095
Область DM	D00000...D32766	D00000...D32767
Область EM, не содержащая банков	E00000...E32766	E00000...E32767
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	—	Только заданные значения
Регистры данных	—	DR0...DR15
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048 +2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(+++) ,(- -)IR0...(- -)IR15	

### Описание

По команде NSRL(583) осуществляется смещение перемещаемых слов D и D+1 вправо на заданное в C количество битов (от бита старшего разряда к биту младшего разряда).

Либо нули, либо значение старшего бита заносится в указанное количество битов вместо перемещаемого слова, начиная с бита старшего разряда.



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда управляющее слово C (количество битов смещения) находится за пределами допустимых пределов. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда результатом сдвига является 0. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении смещения во Флаг переноса (CY) заносится 1. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда при выполнении смещения в старший бит записывается 1. OFF в других случаях.

### Меры предосторожности

Все биты, которые смещаются за пределы указанного слова, теряются, за исключением последнего из утрачиваемых битов, который записывается во флаг переноса (CY)..

Когда количество битов смещения (указанное в C), равно нулю, смещение данных не производится. Тем не менее, соответствующие флаги переводятся в состояние ON или OFF, в зависимости от содержания указанных слов.

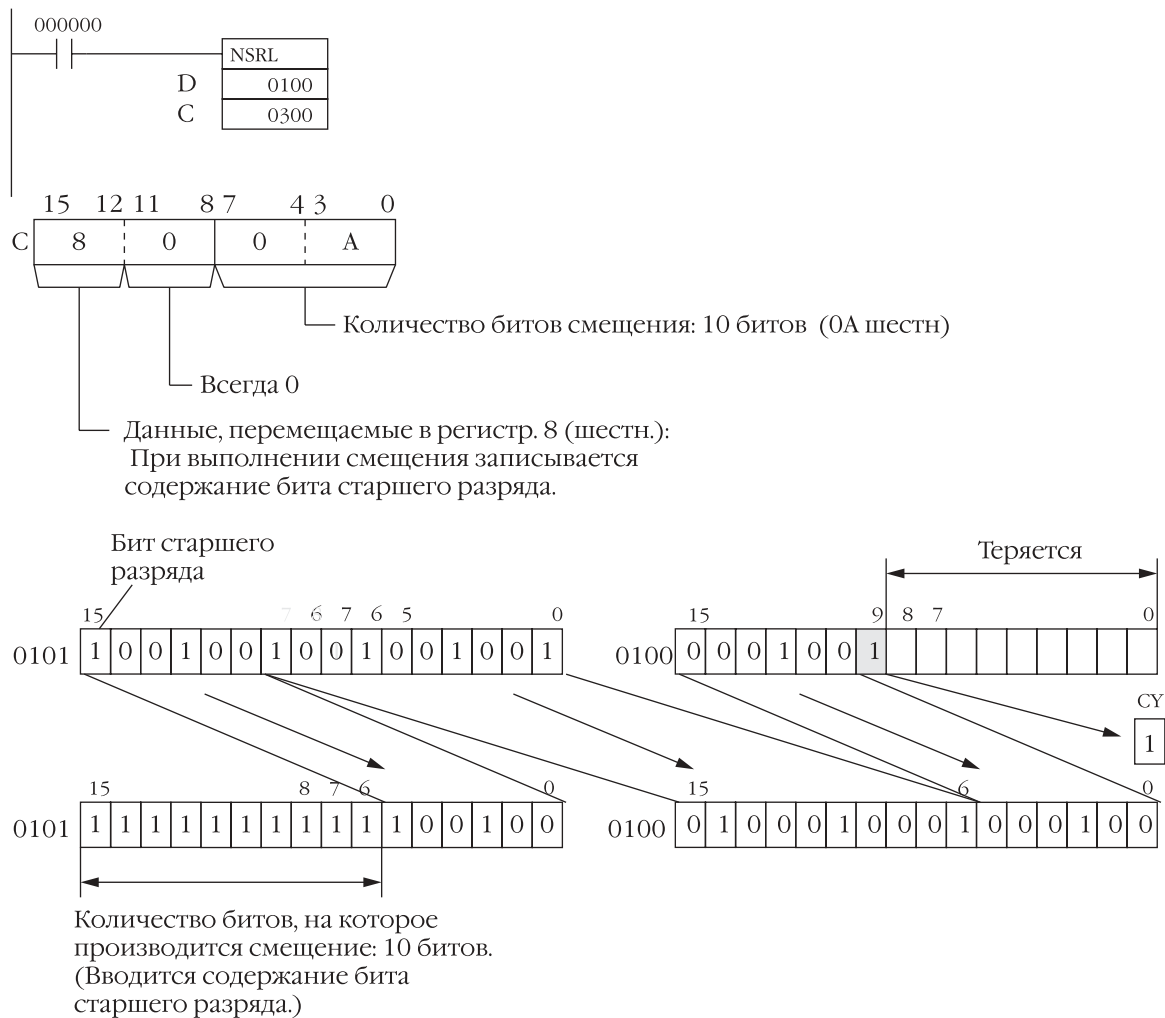
Если содержание управляющего слова C находится за пределами допустимого диапазона, определяется ошибка и флаг ошибки переводится в состояние ON.

В случае, когда в результате выполнения смещения данных содержание D+1 равно 00000000 (шестн.), флаг равенства переводится в состояние ON.

В случае, когда в результате выполнения смещения данных содержание битов старшего разряда D+1 равно 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

Когда бит CIO 000000 переводится в состояние ON, содержание CIO 0100 и CIO 0101 смещается на 10 битов вправо (от бита старшего разряда к биту младшего разряда). Количество битов, на которое производится смещение, указывается в битах 0...7 слова CIO 0300 (данные управления). Содержание бита 15 слова CIO 0100 записывается слева вместо смещенных данных, а содержание последнего из теряемых битов, которые перемещаются за пределы указанного слова, записывается во флаг переноса (CY). Прочие данные теряются.

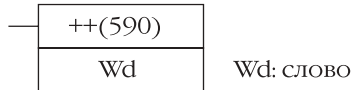


## 3-10 Команды увеличения/уменьшения

## 3-10-1 Команда увеличения двоичного числа INCREMENT BINARY: ++(590)

**Назначение**

По команде ++(590) осуществляется увеличение на единицу содержания указанного слова, четырехзначного числа в шестнадцатеричном коде.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	++(590)
	Выполняется один раз для дифференцирования вверх	@ ++(590)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

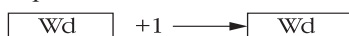
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143
Рабочая область	W000...W511
Область удержания бита (Holding bit area)	H000...H511
Область вспомогательных битов	A448...A959
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область DM	D00000...D32767
Область EM, не содержащая банков	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0C)
Константы	—
Регистры данных	DR0...DR15
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15

**Описание**

По команде ++(590) осуществляется прибавление единицы к содержанию слова Wd, четырехзначному числу в двоичном коде. Заданное слово увеличивается в каждом из циклов, пока условие выполнения команды ++(590) находится в состоянии ON. Когда используется модификация команды с дифференцированием по возрастанию (@++(590)), заданное слово увеличивается только тогда, когда условие выполнения переходит из состояния OFF в состояние ON.



В случае, когда результат равен нулю, флаг равенства переводится в состояние ON. Когда одна из цифр изменяет свое значение из F в 0, флаг переноса переводится в состояние ON. Флаг отрицательного значения переводится в состояние ON, когда в результате прибавления единицы бит 15 слова Wd становится равным 1.

Как флаг равенства, так и флаг переноса переводится в состояние ON, когда содержание Wd изменяет свое значение из FFFF в 0000.

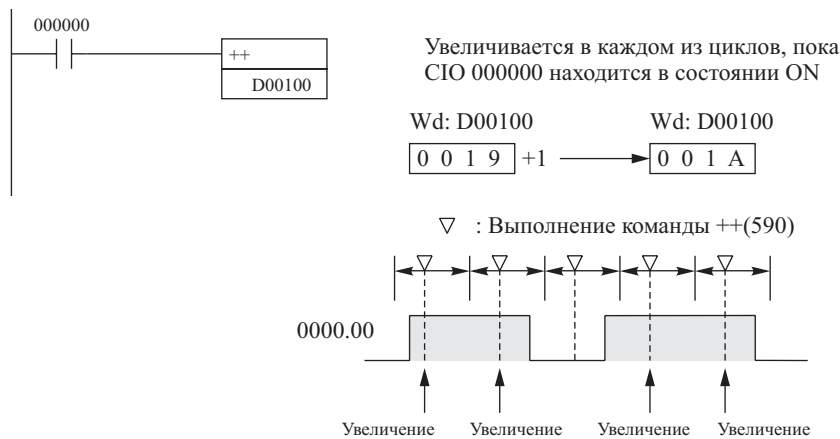
### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды значение Wd равно 0000. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды одна из цифр изменяет свое значение из F в 0. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды бит 15 слова Wd равен 1. OFF в других случаях.

### Примеры

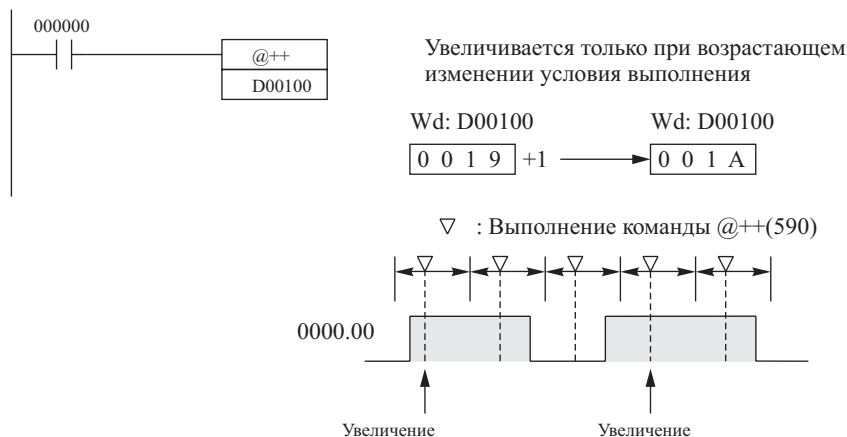
#### Функционирование команды ++(590)

В следующем ниже примере содержание слова D00100 увеличивается на 1 в каждом из циклов, пока CIO 000000 находится в состоянии ON.



#### Функционирование команды @++(590)

В следующем примере применяется дифференцирование вверх, поэтому содержание слова D00100 увеличивается на единицу только тогда, когда CIO000000 переходит из состояния OFF в состояние ON.

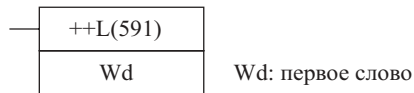


### 3-10-2 Команда увеличения двойного двоичного числа DOUBLE INCREMENT BINARY: ++L(591)

#### Назначение

По команде ++L(591) осуществляется увеличение на единицу содержания заданных слов, восьмизначных чисел в шестнадцатеричном коде.



**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	++L(591)
	Выполняется один раз для дифференцирования вверх	@ ++L(591)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

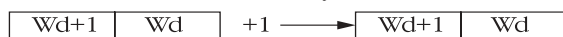
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142
Рабочая область	W000...W510
Область удержания бита (Holding bit area)	H000...H510
Область вспомогательных битов	A448...A958
Область таймера	T0000...T4094
Область счетчика	C0000...C4094
Область DM	D00000...D32766
Область EM, не содержащая банков	E00000...E32766
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = C)
Константы	—
Регистры данных	—
Индексные регистры	IR0...IR15
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(++) ,(- -)IR0...,-(- -)IR15

**Описание**

По команде ++L(591) осуществляется прибавление единицы к содержанию слов Wd+1 и Wd, восьмизначному числу в шестнадцатеричном коде. Содержание заданных слов увеличивается на единицу в каждом из циклов, пока условие выполнения команды ++L(591) находится в состоянии ON. Когда используется модификация команды с дифференцированием по возрастанию (@++L(591)), указанные слова увеличиваются только тогда, когда условие выполнения переходит из состояния OFF в состояние ON.



В случае, когда результат равен 00000000, флаг равенства переводится в состояние ON. Когда одна из цифр изменяет свое значение из F в 0, флаг переноса переводится в состояние ON. Флаг отрицательного значения переводится в состояние ON, когда в результате прибавления бит 15 слова Wd+1 становится равным 1.

Как флаг равенства, так и флаг переноса переводится в состояние ON, когда содержание Wd изменяет свое значение из FFFFFFFF в 00000000.

**Флаги**

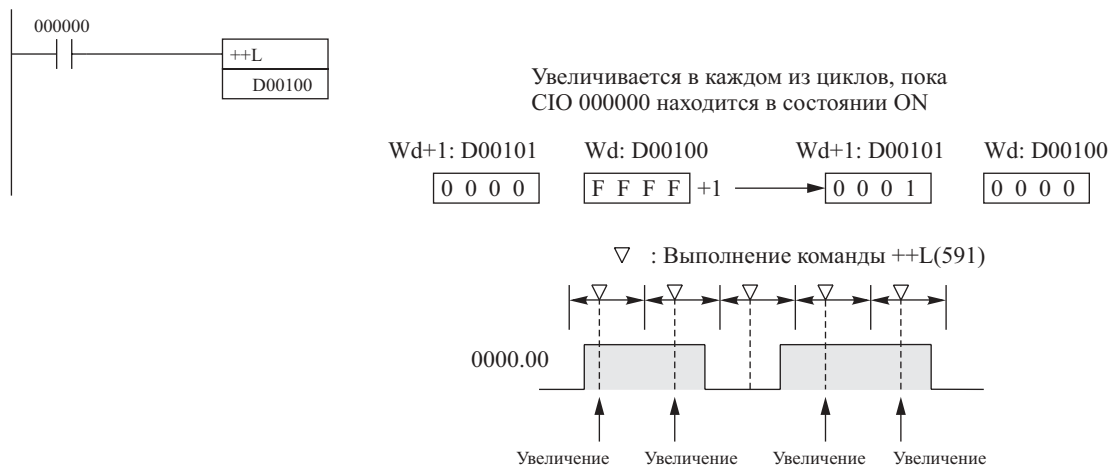
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF

Наименование	Обозначение	Функционирование
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды значение Wd+1 и Wd равно 00000000. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды одна из цифр слов Wd+1 и Wd изменяет свое значение из F в 0. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды бит 15 слова Wd равен 1. OFF в других случаях.

**Примеры**

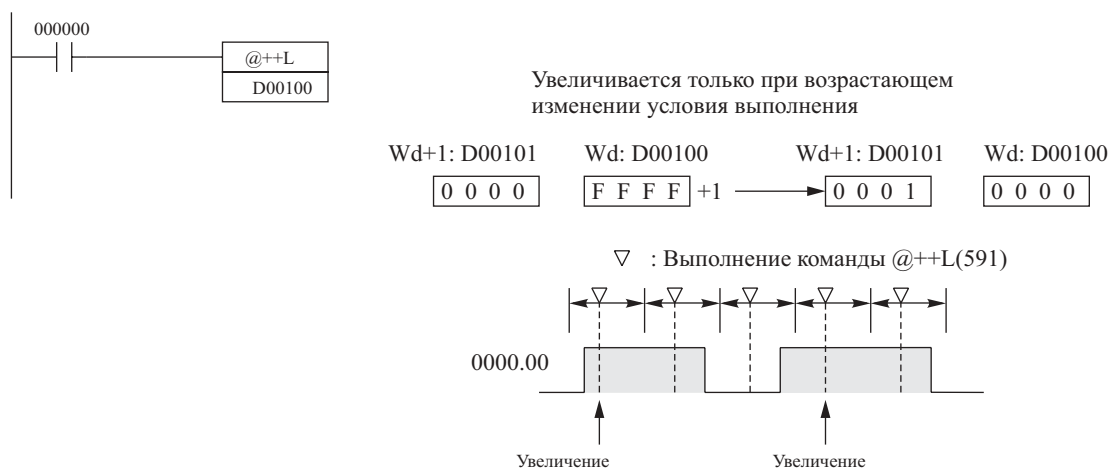
**Функционирование команды ++L(591)**

В следующем ниже примере восьмизначное число, содержащееся в словах D00100 и D00101, увеличивается на 1 в каждом из циклов, пока CIO 000000 находится в состоянии ON.



**Функционирование команды @++L(591)**

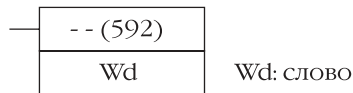
В следующем примере применяется дифференцирование вверх, поэтому содержание слов D00101 и D00100 увеличивается на единицу только тогда, когда CIO000000 переходит из состояния OFF в состояние ON.



**3-10-3 Команда уменьшения двоичного числа DECREMENT BINARY: -- (592)**

**Назначение**

По команде -- (592) осуществляется уменьшение на единицу содержания заданного слова, четырехзначного числа в шестнадцатеричном коде.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	-- (592)
	Выполняется один раз для дифференцирования вверх	@ -- (592)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

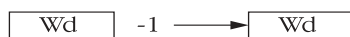
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143
Рабочая область	W000...W511
Область удержания бита (Holding bit area)	H000...H511
Область вспомогательных битов	A448...A959
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область DM	D00000...D32767
Область EM, не содержащая банков	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0C)
Константы	—
Регистры данных	DR0...DR15
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(++) ,(- -)IR0...(- -)IR15

**Описание**

По команде -- (592) осуществляется вычитание единицы из содержания слова Wd в двоичном коде. Заданное слово уменьшается в каждом из циклов, пока условие выполнения команды -- (592) находится в состоянии ON. Когда используется модификация команды с дифференцированием по возрастанию (@--(592)), заданное слово уменьшается только тогда, когда условие выполнения переходит из состояния OFF в состояние ON.



В случае, когда результат равен нулю, флаг равенства переводится в состояние ON. Когда одна из цифр изменяет свое значение из 0 в F, флаг переноса переводится в состояние ON. Флаг отрицательного значения переводится в состояние ON, когда в результате вычитания бит 15 слова Wd становится равным 1. Как флаг равенства, так и флаг отрицательного значения переводится в состояние ON, когда содержание Wd изменяет свое значение из 0000 в FFFF.

**Флаги**

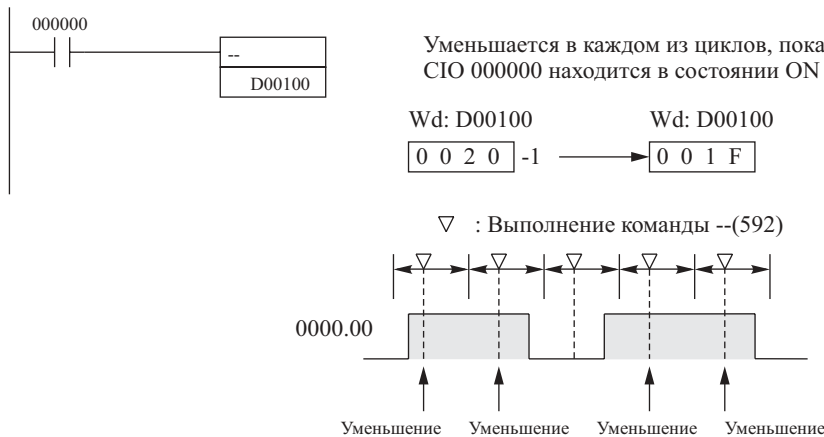
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF

Наименование	Обозначение	Функционирование
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды значение Wd равно 0000. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды одна из цифр изменяет свое значение из 0 в F. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды бит 15 слова Wd равен 1. OFF в других случаях.

**Примеры**

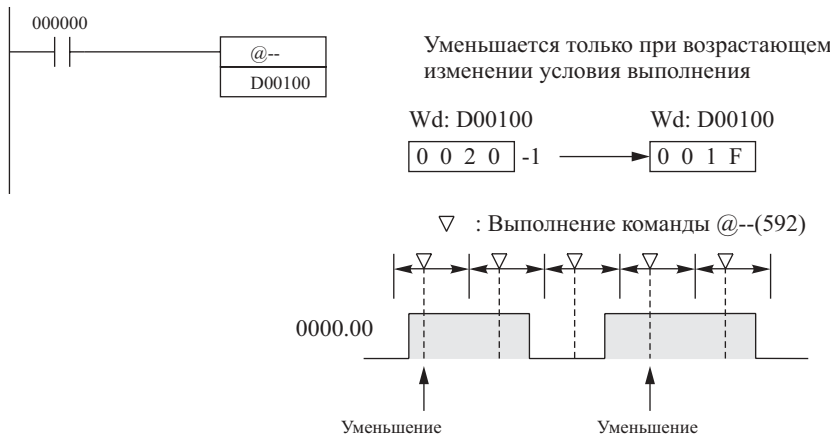
**Функционирование команды -- (592)**

В следующем ниже примере содержание слова D00100 уменьшается на 1 в каждом из циклов, пока CIO 000000 находится в состоянии ON.



**Функционирование команды @-- (592)**

В следующем примере применяется дифференцирование вверх, поэтому содержание слова D00100 уменьшается на единицу только тогда, когда CIO000000 переходит из состояния OFF в состояние ON.



**3-10-4 Команда уменьшения двойного двоичного числа DOUBLE DECREMENT BINARY: -- L(593)**

**Назначение**

По команде -- L(593) осуществляется уменьшение на единицу содержания заданных слов, восьмизначных чисел в шестнадцатеричном коде.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	-- L(593)
	Выполняется один раз для дифференцирования вверх	@ -- L(593)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142
Рабочая область	W000...W510
Область удержания бита (Holding bit area)	H000...H510
Область вспомогательных битов	A448...A958
Область таймера	T0000...T4094
Область счетчика	C0000...C4094
Область DM	D00000...D32766
Область EM, не содержащая банков	E00000...E32766
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0C)
Константы	—
Регистры данных	—
Индексные регистры	IR0...IR15
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15

**Описание**

По команде -- L(593) осуществляется вычитание единицы из содержания слов Wd+1 и Wd, восьмизначного числа в шестнадцатеричном коде. Содержание заданных слов уменьшается на единицу в каждом из циклов, пока условие выполнения команды -- L(593) находится в состоянии ON. Когда используется модификация команды с дифференцированием по возрастанию (@-- L(593)), указанные слова уменьшаются только тогда, когда условие выполнения переходит из состояния OFF в состояние ON.



В случае, когда результат равен 00000000, флаг равенства переводится в состояние ON. Когда одна из цифр изменяет свое значение из 0 в F, флаг переноса переводится в состояние ON. Флаг отрицательного значения переводится в состояние ON, когда в результате прибавления бит 15 слова Wd+1 становится равным 1.

Как флаг равенства, так и флаг отрицательного значения переводится в состояние ON, когда содержание Wd изменяет свое значение из 00000000 в FFFFFFFF.

**Флаги**

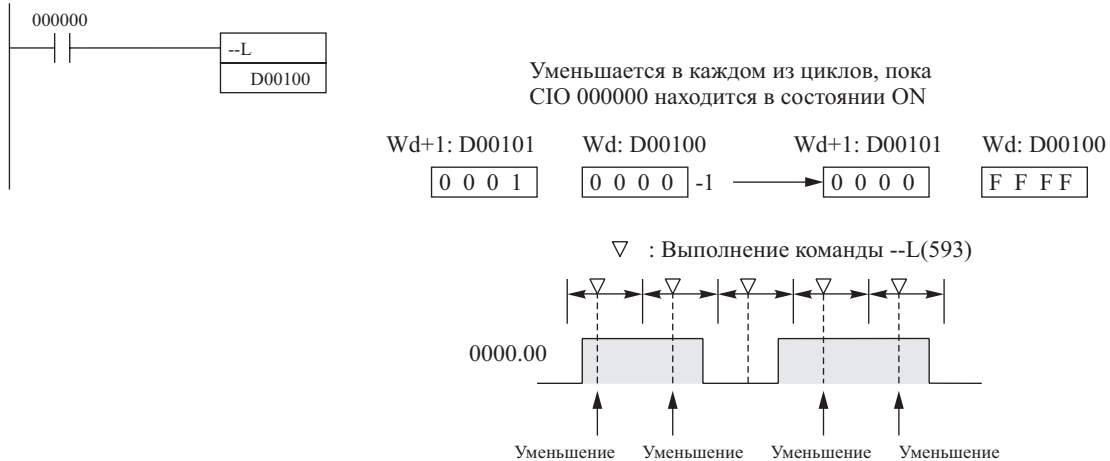
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды значение Wd+1 и Wd равно 00000000. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды одна из цифр слов Wd+1 и Wd изменяет свое значение из 0 в F. OFF в других случаях.

Наименование	Обозначение	Функционирование
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды бит 15 слова Wd равен 1. OFF в других случаях.

**Примеры**

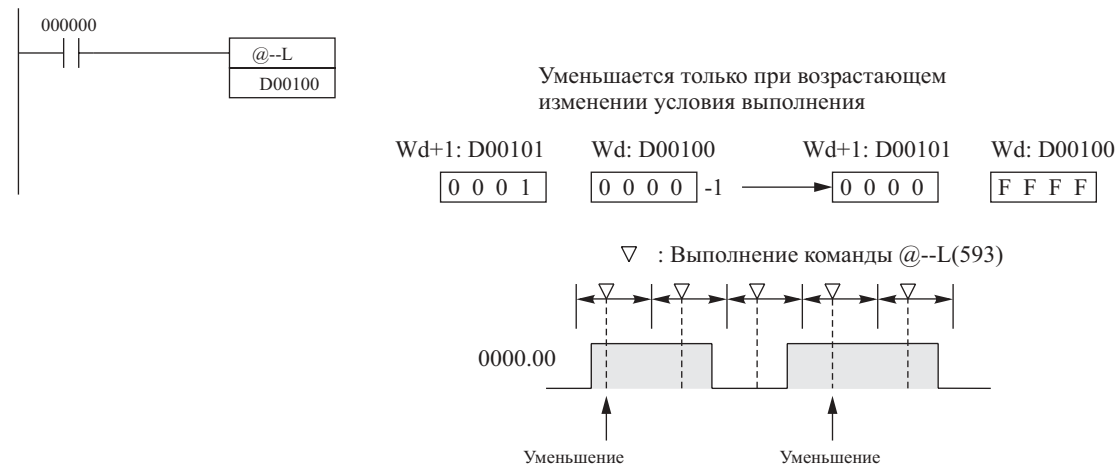
**Функционирование команды --L(593)**

В следующем ниже примере восьмизначное число, содержащееся в словах D00100 и D00101, уменьшается на 1 в каждом из циклов, пока CIO 000000 находится в состоянии ON.



**Функционирование команды @-L(593)**

В следующем примере применяется дифференцирование вверх, поэтому содержание слов D00101 и D00100 уменьшается на единицу только тогда, когда CIO000000 переходит из состояния OFF в состояние ON.

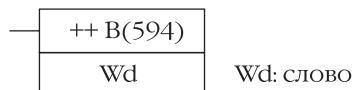


**3-10-5 Команда увеличения двоично-десятичного числа INCREMENT BCD: ++B(594)**

**Назначение**

По команде ++B(594) осуществляется увеличение на единицу содержания заданного слова, четырехзначного числа в двоично-десятичном коде.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	++B(594)
	Выполняется один раз для дифференцирования вверх	@ ++B(594)

	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации	Не поддерживается	

### Применяемые области программы

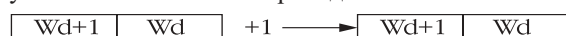
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

### Спецификации операндов

Область	Wd
Область ввода/вывода (область СЮ)	СЮ 0000...СЮ 6143
Рабочая область	W000...W511
Область удержания бита (Holding bit area)	H000...H511
Область вспомогательных битов	A448...A959
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область DM	D00000...D32767
Область EM, не содержащая банков	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0C)
Константы	—
Регистры данных	DR0...DR15
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15

### Описание

По команде ++B(594) осуществляется прибавление единицы к содержанию слова Wd, четырехзначного числа в двоично-десятичном коде. Заданное слово увеличивается в каждом из циклов, пока условие выполнения команды ++B(594) находится в состоянии ON. Когда используется модификация команды с дифференцированием по возрастанию (@++B(594)), заданное слово увеличивается только тогда, когда условие выполнения переходит из состояния OFF в состояние ON.



В случае, когда результат равен нулю (0000), флаг равенства переводится в состояние ON. Когда одна из цифр изменяет свое значение из 9 в 0, флаг переноса переводится в состояние ON.

Как флаг равенства, так и флаг переноса переводится в состояние ON, когда содержание Wd изменяет свое значение из 9999 в 0000.

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание слова WD выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды значение Wd равно 0000. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды одна из цифр изменяет свое значение из 9 в 0. OFF в других случаях.

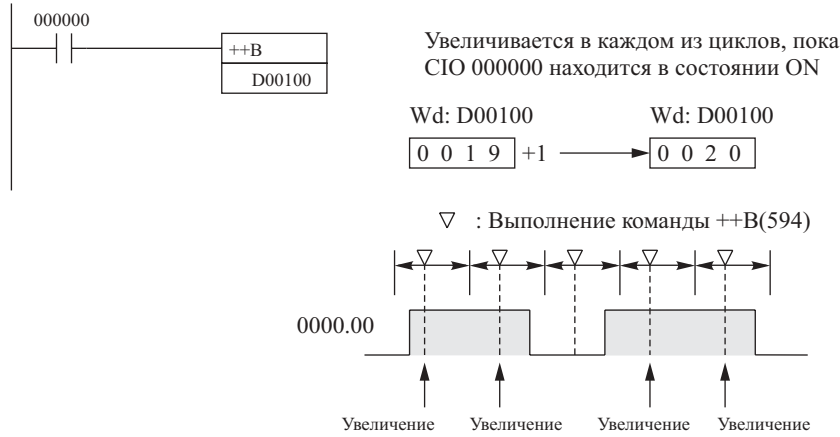
### Меры предосторожности

Содержание слова Wd должно быть числом в двоично-десятичном коде. Если содержание Wd выражено не в двоично-десятичном коде, определяется ошибка, и флаг ошибки переводится в состояние ON.

**Примеры**

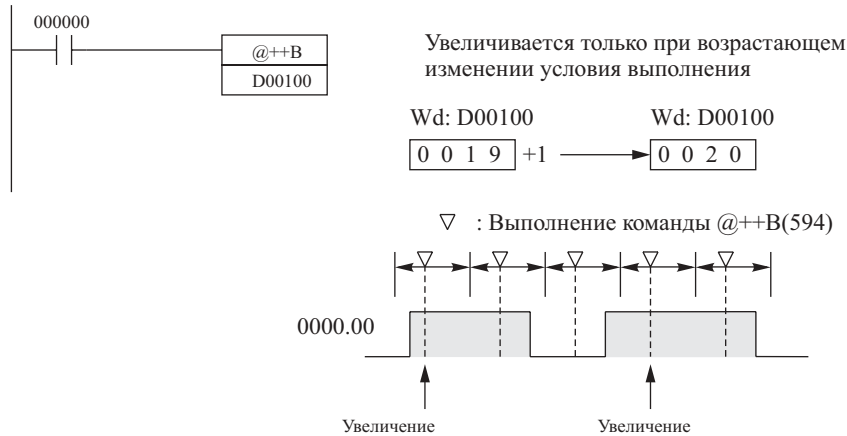
**Функционирование команды ++B(594)**

В следующем ниже примере содержание слова D00100, выраженное в двоично-десятичном коде, увеличивается на 1 в каждом из циклов, пока CIO 000000 находится в состоянии ON.



**Функционирование команды @++B(594)**

В следующем примере применяется дифференцирование вверх, поэтому содержание слова D00100 увеличивается на единицу только тогда, когда CIO000000 переходит из состояния OFF в состояние ON.

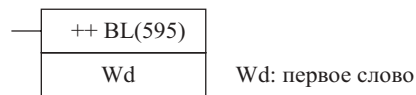


**3-10-6 Команда увеличения двойного двоично-десятичного числа DOUBLE INCREMENT BCD: ++BL(595)**

**Назначение**

По команде ++BL(595) осуществляется увеличение на единицу содержания заданных слов, восьмизначного числа в двоично-десятичном коде.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	++BL(595)
	Выполняется один раз для дифференцирования вверх	@ ++BL(595)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да



### Спецификации операндов

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142
Рабочая область	W000...W510
Область удержания бита (Holding bit area)	H000...H510
Область вспомогательных битов	A448...A958
Область таймера	T0000...T4094
Область счетчика	C0000...C4094
Область DM	D00000...D32766
Область EM, не содержащая банков	E00000...E32766
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0C)
b120Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-)IR15

### Описание

По команде ++BL(595) осуществляется прибавление единицы к содержанию слов Wd+1 и Wd, восьмизначного числа в двоично-десятичном коде. Содержание заданных слов увеличивается на единицу в каждом из циклов, пока условие выполнения команды ++BL(595) находится в состоянии ON. Когда используется модификация команды с дифференцированием по возрастанию (@++BL(595)), указанные слова увеличиваются только тогда, когда условие выполнения переходит из состояния OFF в состояние ON.



В случае, когда результат равен 00000000, флаг равенства переводится в состояние ON. Когда одна из цифр изменяет свое значение из 9 в 0, флаг переноса переводится в состояние ON.

Как флаг равенства, так и флаг переноса переводится в состояние ON, когда содержание Wd изменяет свое значение из 99999999 в 00000000.

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание слов Wd+1 и Wd выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды значение Wd+1 и Wd равно 00000000. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды одна из цифр слов Wd+1 и Wd изменяет свое значение из 9 в 0. OFF в других случаях.

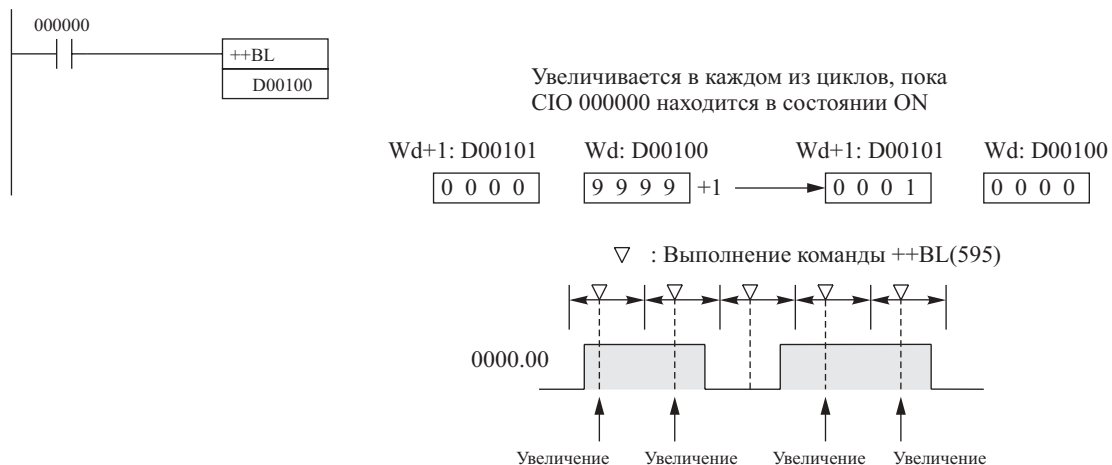
### Меры предосторожности

Содержание слов Wd+1 и Wd должно быть числом в двоично-десятичном коде. Если содержание Wd выражено не в двоично-десятичном коде, определяется ошибка, и флаг ошибки переводится в состояние ON.

### Примеры

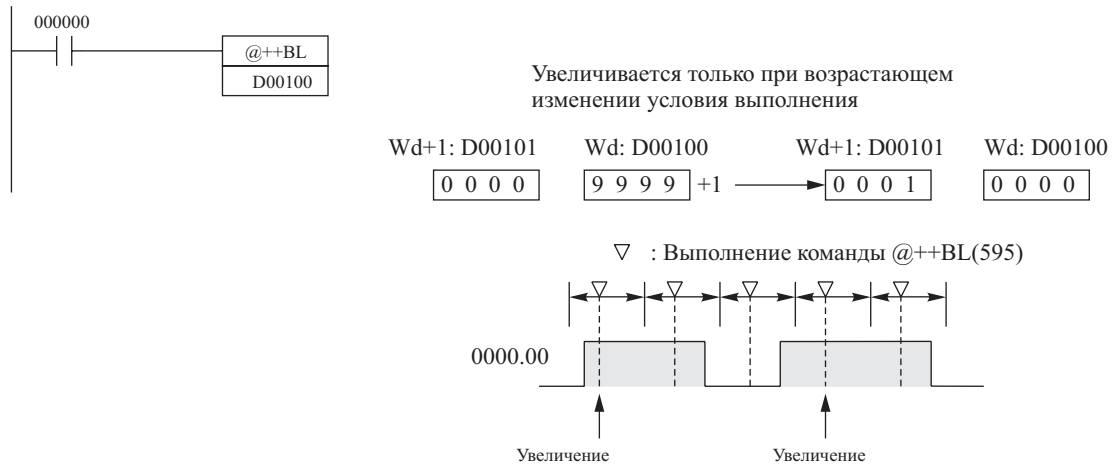
#### Функционирование команды ++BL(595)

В следующем ниже примере восьмизначное число, выраженное в двоично-десятичном коде и содержащееся в словах D00100 и D00101, увеличивается на 1 в каждом из циклов, пока CIO 000000 находится в состоянии ON.



**Функционирование команды @++BL(595)**

В следующем примере применяется дифференцирование вверх, поэтому содержание слов D00101 и D00100 увеличивается на единицу только тогда, когда СЮ000000 переходит из состояния OFF в состояние ON.

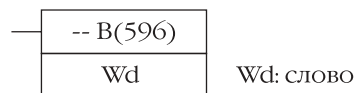


**3-10-7 Команда уменьшения двоично-десятичного числа DECREMENT BCD: --B(596)**

**Назначение**

По команде --B(596) осуществляется уменьшение на единицу содержания заданного слова, четырехзначного числа в двоично-десятичном коде.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	--B(596)
	Выполняется один раз для дифференцирования вверх	@ --B(596)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

### Спецификации операндов

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143
Рабочая область	W000...W511
Область удержания бита (Holding bit area)	H000...H511
Область вспомогательных битов	A448...A959
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область DM	D00000...D32767
Область EM, не содержащая банков	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0C)
Константы	–
Регистры данных	DR0...DR15
Индексные регистры	–
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,(- -)IR0...(- -)IR15

### Описание

По команде --В(596) осуществляется вычитание единицы из содержания слова Wd, четырехзначного числа в двоично-десятичном коде. Заданное слово уменьшается в каждом из циклов, пока условие выполнения команды --В(596) находится в состоянии ON. Когда используется модификация команды с дифференцированием по возрастанию (@--В(596)), содержание заданного слова уменьшается только тогда, когда условие выполнения переходит из состояния OFF в состояние ON.



В случае, когда результат равен нулю (0000), флаг равенства переводится в состояние ON. Когда одна из цифр изменяет свое значение из 0 в 9, флаг переноса переводится в состояние ON.

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание слова WD выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды значение Wd равно 0000. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды одна из цифр изменяет свое значение из 0 в 9. OFF в других случаях.

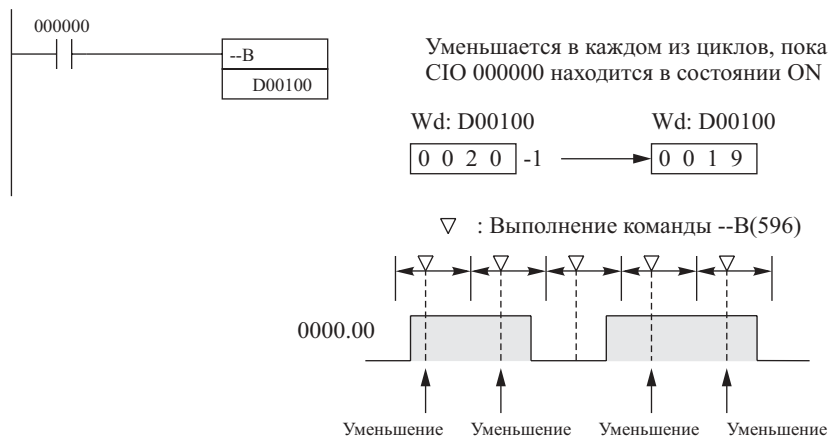
### Меры предосторожности

Содержание слова Wd должно быть числом в двоично-десятичном коде. Если содержание Wd выражено не в двоично-десятичном коде, определяется ошибка, и флаг ошибки переводится в состояние ON.

### Примеры

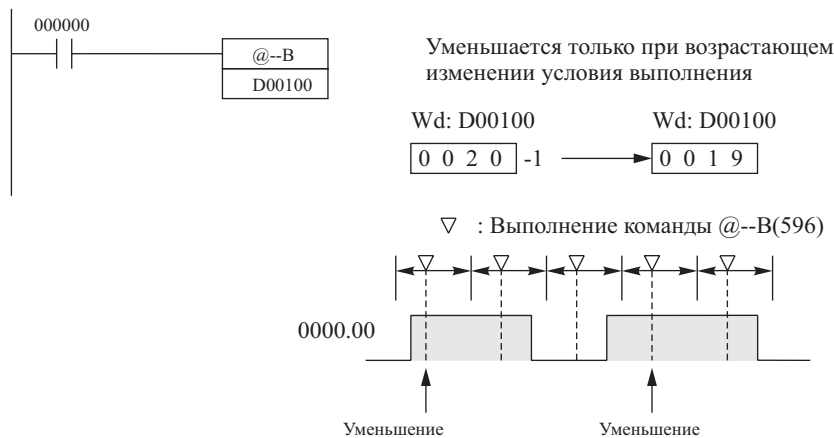
#### Функционирование команды --В(596)

В следующем ниже примере содержание слова D00100, выраженное в двоично-десятичном коде, уменьшается на 1 в каждом из циклов, пока CIO 000000 находится в состоянии ON.



**Функционирование команды @--B(596)**

В следующем примере применяется дифференцирование вверх, поэтому содержание слова D00100 уменьшается на единицу только тогда, когда СЮ 000000 переходит из состояния OFF в состояние ON.

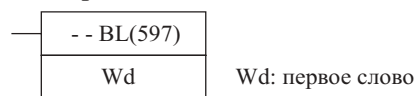


**3-10-8 Команда уменьшения двойного двоично-десятичного числа DOUBLE DECEMENT BCD: --BL(597)**

**Назначение**

По команде --BL(597) осуществляется уменьшение на единицу содержания заданных слов, восьмизначного числа в двоично-десятичном коде.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON.	--BL(597)
	Выполняется один раз для дифференцирования вверх	@ --BL(597)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается.
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

### Спецификации операндов

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142
Рабочая область	W000...W510
Область удержания бита (Holding bit area)	H000...H510
Область вспомогательных битов	A448...A958
Область таймера	T0000...T4094
Область счетчика	C0000...C4094
Область DM	D00000...D32766
Область EM, не содержащая банков	E00000...E32766
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0C)
Константы	–
Регистры данных	–
Индексные регистры	–
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-)IR15

### Описание

По команде --BL(597) осуществляется вычитание единицы из содержания слов Wd+1 и Wd, восьмизначного числа, выраженного в двоично-десятичном коде. Содержание заданных слов уменьшается на единицу в каждом из циклов, пока условие выполнения команды --BL(597) находится в состоянии ON. Когда используется модификация команды с дифференцированием по возрастанию (@--BL(597)), содержание указанных слов уменьшается только тогда, когда условие выполнения переходит из состояния OFF в состояние ON.



В случае, когда результат равен 00000000, флаг равенства переводится в состояние ON. Когда одна из цифр изменяет свое значение из 0 в 9, флаг переноса переводится в состояние ON.

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание слов Wd+1 и Wd выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды значение Wd+1 и Wd равно 00000000. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды одна из цифр слов Wd+1 и Wd изменяет свое значение из 0 в 9. OFF в других случаях.

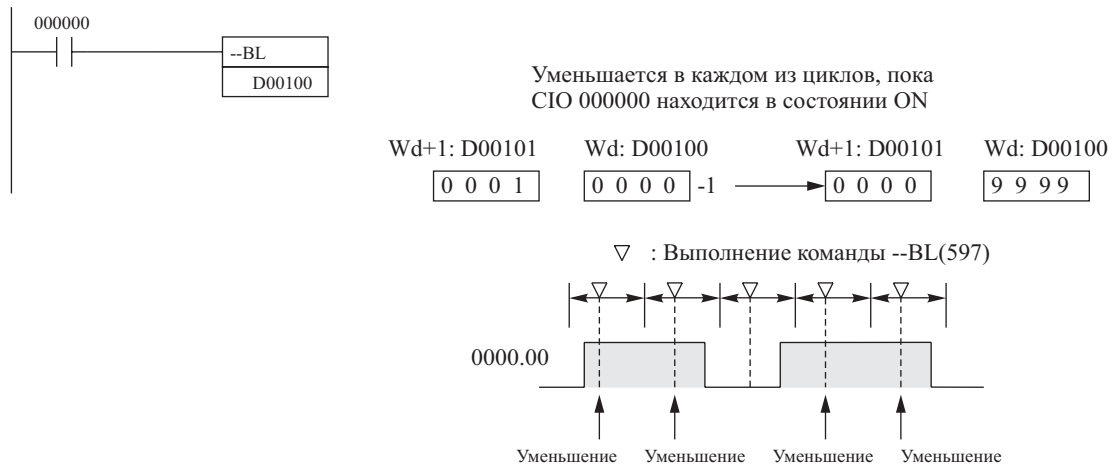
### Меры предосторожности

Содержание слов Wd+1 и Wd должно быть числом в двоично-десятичном коде. Если содержание Wd выражено не в двоично-десятичном коде, определяется ошибка, и флаг ошибки переводится в состояние ON.

### Примеры

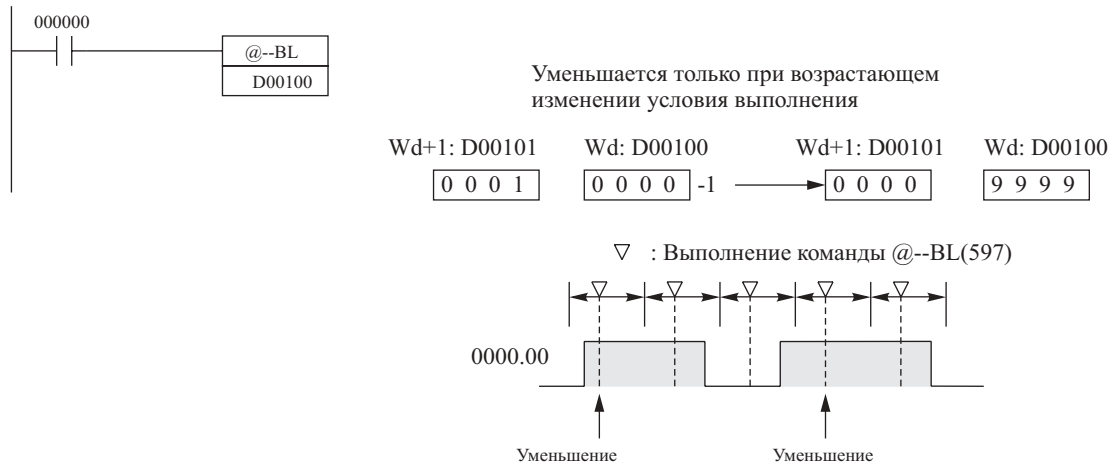
#### Функционирование команды --BL(597)

В следующем ниже примере восьмизначное число, выраженное в двоично-десятичном коде и содержащееся в словах D00100 и D00101, уменьшается на 1 в каждом из циклов, пока CIO 000000 находится в состоянии ON.



**Функционирование команды @--BL(597)**

В следующем примере применяется дифференцирование вверх, поэтому содержание слов D00101 и D00100 уменьшается на единицу только тогда, когда CIO000000 переходит из состояния OFF в состояние ON.



## 3-11 Символьные математические команды

В настоящем разделе приводится описание символьных математических команд, которые выполняют арифметические операции с двоично-десятичными или двоичными данными.

Команда	Мнемоническое изображение	Код функции	Страница
SIGNED BINARY ADD WITHOUT CARRY	+	400	288
DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+L	401	290
SIGNED BINARY ADD WITH CARRY	+C	402	292
DOUBLE SIGNED BINARY ADD WITH CARRY	+CL	403	294
BCD ADD WITHOUT CARRY	+B	404	295
DOUBLE BCD ADD WITHOUT CARRY	+BL	405	297
BCD ADD WITH CARRY	+BC	406	299
DOUBLE BCD ADD WITH CARRY	+BCL	407	300
SIGNED BINARY SUBTRACT WITHOUT CARRY	-	410	302
DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	-L	411	304
SIGNED BINARY SUBTRACT WITH CARRY	-C	412	307
DOUBLE SIGNED BINARY SUBTRACT WITH CARRY	-CL	413	309
BCD SUBTRACT WITHOUT CARRY	-B	414	312
DOUBLE BCD SUBTRACT WITHOUT CARRY	-BL	415	313
BCD SUBTRACT WITH CARRY	-BC	416	316
DOUBLE BCD SUBTRACT WITH CARRY	-BCL	417	318
SIGNED BINARY MULTIPLY	*	420	320
DOUBLE SIGNED BINARY MULTIPLY	*L	421	321
UNSIGNED BINARY MULTIPLY	*U	422	323
DOUBLE UNSIGNED BINARY MULTIPLY	*UL	423	324
BCD MULTIPLY	*B	424	326
DOUBLE BCD MULTIPLY	*BL	425	327
SIGNED BINARY DIVIDE	/	430	329
DOUBLE SIGNED BINARY DIVIDE	/L	431	330
UNSIGNED BINARY DIVIDE	/U	432	332
DOUBLE UNSIGNED BINARY DIVIDE	/UL	433	334
BCD DIVIDE	/B	434	335
DOUBLE BCD DIVIDE	/BL	435	337

## 3-11-1 Команда сложения без переноса двоичных данных со знаком SIGNED BINARY ADD WITHOUT CARRY: +(400)

**Назначение**

По команде +(400) осуществляется суммирование четырехзначных шестнадцатеричных данных (простого слова) и/или констант.

**Символ релейно-контактной схемы**

+(400)	
Au	Au: Первое слагаемое
Ad	Ad: Второе слагаемое
R	R: Результирующее слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	+(400)
	Выполняется один раз для дифференцирования вверх	@ +(400)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Au	Ad	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF	-	
Регистры данных	DR0...DR15		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047,IR0 -2048...+2047,IR15 DR0...DR15,IR0...IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15		

**Описание**

По команде +(400) осуществляется сложение двоичных значений чисел, находящихся в Au и Ad и вывод результата в слово R.

$$\boxed{\text{Au}} \quad (\text{Двоичное со знаком})$$

$$+ \quad \boxed{\text{Ad}} \quad (\text{Двоичное со знаком})$$

Флаг переноса CY переводится в состояние ON, когда в результате сложения выполняется перенос

$$\boxed{\text{CY}} \quad \boxed{\text{R}} \quad (\text{Двоичное со знаком})$$

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды осуществляется перенос. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, когда результат сложения двух положительных чисел находится в пределах от 8000 до FFFF (шестн.). OFF в других случаях.
Флаг отрицательного переполнения	UF	Переводится в состояние ON, когда результат сложения двух отрицательных чисел находится в пределах от 0000 до 7FFF (шестн.). OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда в результате сложения бит старшего разряда результирующего слова равен 1. OFF в других случаях.

**Меры предосторожности**

После выполнения команды +(400) флаг ошибки переводится в состояние OFF.



Когда в результате выполнения команды результирующее слово R равно 0000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате сложения выполняется перенос, флаг переноса переводится в состояние ON.

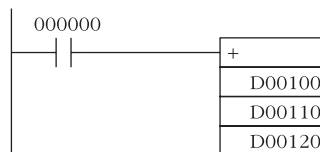
Когда в результате сложения двух положительных чисел сумма является отрицательным числом (в пределах от 8000 до FFFF шестн.), флаг переполнения переводится в состояние ON.

Когда в результате сложения двух отрицательных чисел сумма является положительным числом (в пределах от 0000 до 7FFF шестн.) флаг отрицательного переполнения переводится в состояние ON.

Если в результате выполнения команды содержание старшего бита слова R равно 1, флаг отрицательного значения переводится в состояние ON.

### Примеры

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится сложение двоичных значений четырехзначных чисел со знаком содержащихся в словах D00100 и D00110, и результат записывается в D00120.

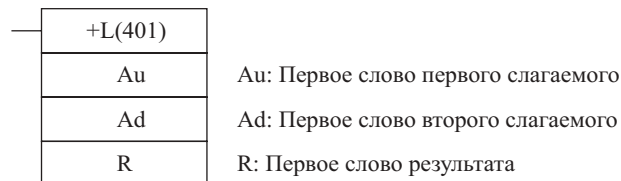


### 3-11-2 Команда сложения без переноса двойных двоичных данных со знаком DOUBLE SIGNED BINARY ADD WITHOUT CARRY: +L(401)

#### Назначение

По команде +L(401) осуществляется суммирование восьмизначных шестнадцатеричных данных (двойного слова) и/или констант.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	+L(401)
	Выполняется один раз для дифференцирования вверх	@ +L(401)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Спецификации операндов

Область	Au	Ad	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142		
Рабочая область	W000...W510		
Область удержания бита (Holding bit area)	H000...H510		
Область вспомогательных битов	A000...A958	A448...A958	
Область таймера	T0000...T4094		
Область счетчика	C0000...C4094		
Область DM	D00000...D32766		
Область EM, не содержащая банков	E00000...E32766		
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		

Область	Au	Ad	R
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#00000000...#FFFF FFFF (двоичные)	–	
Регистры данных	–		
Индексные регистры	IR0...IR15		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)... ,IR15+(++) ,(- -)IR0...,(- -)IR15		

**Описание**

По команде +L(401) осуществляется сложение двоичных значений чисел, находящихся в Au, Au+1 и Ad, Ad+1, и вывод результата в слова R и R+1.

$$\begin{array}{r}
 \boxed{\text{Au}+1} \quad \boxed{\text{Au}} \quad (\text{Двоичное со знаком}) \\
 + \\
 \boxed{\text{Ad}+1} \quad \boxed{\text{Ad}} \quad (\text{Двоичное со знаком}) \\
 \hline
 \end{array}$$

Флаг переноса CY переводится в состояние ON, когда в результате сложения выполняется перенос

$$\boxed{\text{CY}} \quad \boxed{\text{R}+1} \quad \boxed{\text{R}} \quad (\text{Двоичное со знаком})$$

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды осуществляется перенос. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, когда результат сложения двух положительных чисел находится в пределах от 80000000 до FFFFFFFF (шестн.). OFF в других случаях.
Флаг отрицательного переполнения	UF	Переводится в состояние ON, когда результат сложения двух отрицательных чисел находится в пределах от 00000000 до 7FFFFFFF (шестн.). OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда в результате сложения бит старшего разряда результирующего слова равен 1. OFF в других случаях.

**Меры предосторожности**

После выполнения команды +L(401) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения команды содержание результирующих слов R и R+1 равно 00000000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате сложения выполняется перенос, флаг переноса переводится в состояние ON.

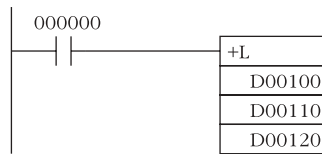
Когда в результате сложения двух положительных чисел сумма является отрицательным числом (в пределах от 80000000 до FFFFFFFF шестн.), флаг переполнения переводится в состояние ON.

Когда в результате сложения двух отрицательных чисел сумма является положительным числом (в пределах от 00000000 до 7FFFFFFF шестн.) флаг отрицательного переполнения переводится в состояние ON.

Если в результате выполнения команды содержание старшего бита слова R+1 равно 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится сложение содержания слов D00100, D00101 и D00110, D00111 в виде двоичных значений восьмизначных чисел со знаком, и результат записывается в D00120 и D00121.

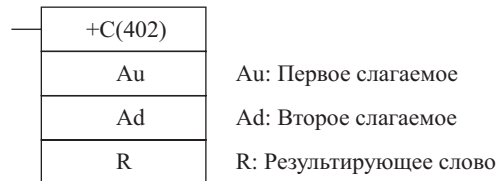


### 3-11-3 Команда сложения двоичных данных со знаком с переносом SIGNED BINARY ADD WITH CARRY: +C(402)

#### Назначение

По команде +C(402) осуществляется суммирование четырехзначных шестнадцатеричных данных (простого слова) и/или констант с использованием флага переноса.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	+C(402)
	Выполняется один раз для дифференцирования вверх	@ +C(402)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Спецификации операндов

Область	Au	Ad	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF (двоичные)	-	
Регистры данных	DR0...DR15		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(-)IR0...,(-)IR15		

**Описание**

По команде +C(402) осуществляется сложение двоичных значений чисел, находящихся в Au, Ad, и флага переноса CY, а также вывод результата в слово R.

(Двоичное со знаком)

(Двоичное со знаком)

+

Флаг переноса CY переводится в состояние ON, когда при выполнении команды выполняется перенос

(Двоичное со знаком)

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды осуществляется перенос. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, когда результат сложения двух положительных чисел и флага переноса находится в пределах от 8000 до FFFF (шестн.). OFF в других случаях.
Флаг отрицательного переполнения	UF	Переводится в состояние ON, когда результат сложения двух отрицательных чисел и флага переноса находится в пределах от 0000 до 7FFF (шестн.). OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда в результате сложения бит старшего разряда результирующего слова равен 1. OFF в других случаях.

**Меры предосторожности**

После выполнения команды +C(402) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения команды результирующее слово R равно 0000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате сложения выполняется перенос, флаг переноса переводится в состояние ON.

Когда в результате сложения двух положительных чисел и флага переноса сумма является отрицательным числом (в пределах от 8000 до FFFF шестн.), флаг переполнения переводится в состояние ON.

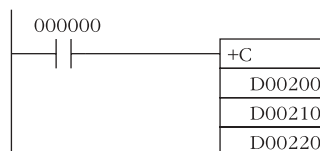
Когда в результате сложения двух отрицательных чисел и флага переноса сумма является положительным числом (в пределах от 0000 до 7FFF шестн.) флаг отрицательного переполнения переводится в состояние ON.

Если в результате выполнения команды содержание старшего бита слова R равно 1, флаг отрицательного значения переводится в состояние ON.

**Примечание:** Для сброса Флага переноса (CY) выполняйте команду (CLC041).

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится сложение содержания слов D00200, D00210 и CY в виде двоичных значений четырехзначных чисел со знаком, и результат записывается в D00220.



### 3-11-4 Команда сложения двойных двоичных данных со знаком с переносом DOUBLE SIGNED BINARY ADD WITH CARRY: +CL(403)

#### Назначение

По команде +CL(403) осуществляется суммирование восьмизначных шестнадцатеричных данных (двойного слова) и/или констант с использованием флага переноса.

#### Символ релейно-контактной схемы

+CL(403)	
Au	Au: Первое слово первого слагаемого
Ad	Ad: Первое слово второго слагаемого
R	R: Первое слово результата

#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	+CL(403)
	Выполняется один раз для дифференцирования вверх	@ +CL(403)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

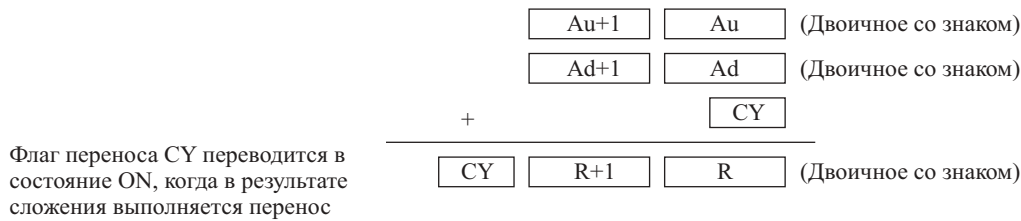
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Спецификации операндов

Область	Au	Ad	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142		
Рабочая область	W000...W510		
Область удержания бита (Holding bit area)	H000...H510		
Область вспомогательных битов	A000...A958	A448...A958	
Область таймера	T0000...T4094		
Область счетчика	C0000...C4094		
Область DM	D00000...D32766		
Область EM, не содержащая банков	E00000...E32766		
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#00000000...#FFFF FFFF (двоичные)	–	
Регистры данных	–		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15		

#### Описание

По команде +CL(403) осуществляется сложение двоичных чисел, находящихся в Au, Au+1 и Ad, Ad+1 и флага переноса, а также вывод результата в слова R и R+1.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды осуществляется перенос. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, когда результат сложения двух положительных чисел и флага переноса находится в пределах от 80000000 до FFFFFFFF (шестн.). OFF в других случаях.
Флаг отрицательного переполнения	UF	Переводится в состояние ON, когда результат сложения двух отрицательных чисел и флага переноса находится в пределах от 00000000 до 7FFFFFFF (шестн.). OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда в результате сложения бит старшего разряда результирующего слова равен 1. OFF в других случаях.

**Меры предосторожности**

После выполнения команды +CL(403) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения команды содержание результирующих слов R и R+1 равно 00000000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате сложения выполняется перенос, флаг переноса переводится в состояние ON.

Когда в результате сложения двух положительных чисел и флага переноса сумма является отрицательным числом (в пределах от 80000000 до FFFFFFFF шестн.), флаг переполнения переводится в состояние ON.

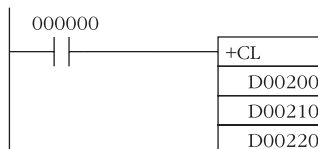
Когда в результате сложения двух отрицательных чисел и флага переноса сумма является положительным числом (в пределах от 00000000 до 7FFFFFFF шестн.) флаг отрицательного переполнения переводится в состояние ON.

Если в результате выполнения команды содержание старшего бита слова R+1 равно 1, флаг отрицательного значения переводится в состояние ON.

**Примечание:** Для сброса Флага переноса (CY) выполняйте команду (CLC041).

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится сложение содержания слов D00201, D00200 и D00211, D00210 в виде двоичных значений восьмизначных двоичных чисел со знаком, и флага переноса. Результат записывается в D00221 и D00220.

**3-11-5 Команда сложения без переноса двоично-десятичных данных BCD ADD WITHOUT CARRY: +B(404)****Назначение**

По команде +B(404) осуществляется суммирование четырехзначных двоично-десятичных данных (простого слова) и/или констант.

**Символ релейно-контактной схемы**

+B(404)	
Au	Au: Первое слагаемое
Ad	Ad: Второе слагаемое
R	R: Результирующее слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	+B(404)
	Выполняется один раз для дифференцирования вверх	@ +B(404)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Au	Ad	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	0000...9999 (двоично-десятичные)	-	
Регистры данных	DR0...DR15		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15		

**Описание**

По команде +B(404) осуществляется сложение двоично-десятичных чисел, находящихся в Au и Ad, и вывод результата в слово R.

$$\boxed{\text{Au}} \text{ (Двоично-десятичное)} \\ + \boxed{\text{Ad}} \text{ (Двоично-десятичное)}$$

Флаг переноса CY переводится в состояние ON, когда в результате сложения выполняется перенос

$$\boxed{\text{CY}} \quad \boxed{\text{R}} \text{ (Двоично-десятичное)}$$

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание Au выражено не в двоично-десятичном коде. Переводится в состояние ON, когда содержание Ad выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды осуществляется перенос. OFF в других случаях.

**Меры предосторожности**

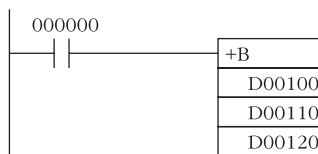
Если содержание слов Au и Ad выражено не в двоично-десятичном коде, определяется ошибка и флаг ошибки переводится в состояние ON.

Когда в результате выполнения команды результирующее слово R равно 0000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате сложения выполняется перенос, флаг переноса переводится в состояние ON.

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится сложение содержания слов D00100 и D00110, в виде четырехзначных двоично-десятичных чисел, и результат записывается в D00120.

**3-11-6 Команда сложения без переноса двойных двоично-десятичных данных DOUBLE BCD ADD WITHOUT CARRY: +BL(405)****Назначение**

По команде +BL(405) осуществляется суммирование восьмизначных двоично-десятичных данных (двойного слова) и/или констант.

**Символ релейно-контактной схемы**

+BL(405)	
Au	Au: Первое слово первого слагаемого
Ad	Ad: Первое слово второго слагаемого
R	R: Первое слово результата

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	+BL(405)
	Выполняется один раз для дифференцирования вверх	@ +BL(405)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Au	Ad	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142		



Область	Au	Ad	R
Рабочая область	W000...W510		
Область удержания бита (Holding bit area)	H000...H510		
Область вспомогательных битов	A000...A958	A448...A958	
Область таймера	T0000...T4094		
Область счетчика	C0000...C4094		
Область DM	D00000...D32766		
Область EM, не содержащая банков	E00000...E32766		
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#00000000...#99999999 (двоично-десятичные)	-	
Регистры данных	-		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15		

**Описание**

По команде +L(401) осуществляется сложение двоично-десятичных чисел, находящихся в Au, Au+1 и Ad, Ad+1 и вывод результата в слова R и R+1.

$$\begin{array}{r}
 \boxed{\text{Au}+1} \quad \boxed{\text{Au}} \quad (\text{Двоично-десятичные}) \\
 + \\
 \boxed{\text{Ad}+1} \quad \boxed{\text{Ad}} \quad (\text{Двоично-десятичные}) \\
 \hline
 \end{array}$$

Флаг переноса CY переводится в состояние ON, когда в результате сложения выполняется перенос

$$\begin{array}{r}
 \boxed{\text{CY}} \quad \boxed{\text{R}+1} \quad \boxed{\text{R}} \quad (\text{Двоично-десятичные})
 \end{array}$$

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание Au и Au+1 выражено не в двоично-десятичном коде. Переводится в состояние ON, когда содержание Ad и Ad+1 выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды осуществляется перенос. OFF в других случаях.

**Меры предосторожности**

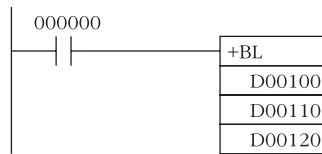
Если содержание слов Au, Au+1 и Ad, Ad+1 выражено не в двоично-десятичном коде, определяется ошибка и флаг ошибки переводится в состояние ON.

Когда в результате выполнения команды содержание R и R+1 равно 00000000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате сложения выполняется перенос, флаг переноса переводится в состояние ON.

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится суммирование содержания слов D00101, D00100 и D00111, D00110 в виде восьмизначных двоично-десятичных чисел, и результат записывается в D00121 и D00120.

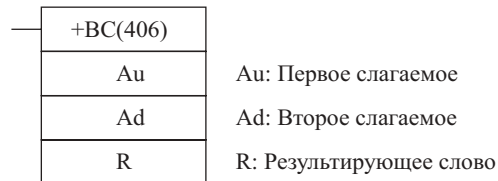


### 3-11-7 Команда сложения двоично-десятичных данных с переносом BCD ADD WITH CARRY: +BC(406)

#### Назначение

По команде +BC(406) осуществляется суммирование четырехзначных двоично-десятичных данных (простого слова) и/или констант с использованием флага переноса (CY).

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	+BC(406)
	Выполняется один раз для дифференцирования вверх	@ +BC(406)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Спецификации операндов

Область	Au	Ad	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...9999 (двоично-десятичные)	-	
Регистры данных	DR0...DR15		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,(- -)IR0...(- -)IR15		

**Описание**

По команде +BC(406) осуществляется сложение двоично-десятичных чисел, находящихся в Au и Ad, и флага переноса, а также вывод результата в слово R.

Au (Двоично-десятичное)

Ad (Двоично-десятичное)

+ CY

Флаг переноса CY переводится в состояние ON, когда в результате вложения выполняется перенос

CY R (Двоично-десятичное)

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание Au выражено не в двоично-десятичном коде. Переводится в состояние ON, когда содержание Ad выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды осуществляется перенос. OFF в других случаях.

**Меры предосторожности**

Если содержание слов Au и Ad выражено не в двоично-десятичном коде, определяется ошибка и флаг ошибки переводится в состояние ON.

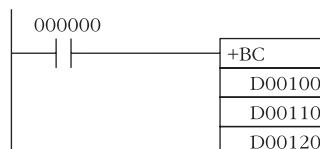
Когда в результате выполнения команды результирующее слово R равно 0000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате сложения выполняется перенос, флаг переноса переводится в состояние ON.

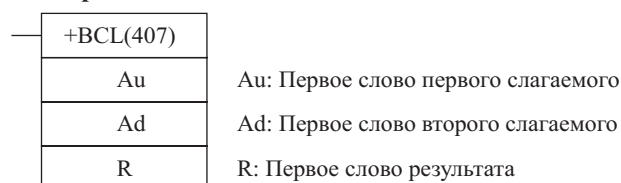
**Примечание:** Для сброса Флага переноса (CY) выполняйте команду (CLC041).

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится сложение содержания слов D00100 и D00110, в виде четырехзначных двоично-десятичных чисел, и флага переноса. Результат записывается в D00120.

**3-11-8 Команда сложения двойных двоично-десятичных данных с переносом DOUBLE BCD ADD WITH CARRY: +BCL(407)****Назначение**

По команде +BCL(407) осуществляется суммирование восьмизначных двоично-десятичных данных (двойного слова) и/или констант с использованием флага переноса (CY).

**Символ релейно-контактной схемы**

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	+BCL(407)
	Выполняется один раз для дифференцирования вверх	@ +BCL(407)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Au	Ad	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142		
Рабочая область	W000...W510		
Область удержания бита (Holding bit area)	H000...H510		
Область вспомогательных битов	A000...A958	A448...A958	
Область таймера	T0000...T4094		
Область счетчика	C0000...C4094		
Область DM	D00000...D32766		
Область EM, не содержащая банков	E00000...E32766		
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#00000000...#99999999 (двоично-десятичные)	—	
Регистры данных	—		
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-(-)IR15		

**Описание**

По команде +BCL(407) осуществляется сложение двоично-десятичных чисел, находящихся в Au, Au+1 и Ad, Ad+1, и флага переноса. Результат выводится в слова R и R+1.

Au+1	Au	(Двоично-десятичные)
------	----	----------------------

Ad+1	Ad	(Двоично-десятичные)
------	----	----------------------

+

CY

Флаг переноса CY переводится в состояние ON, когда в результате сложения выполняется перенос

CY	R+1	R	(Двоично-десятичные)
----	-----	---	----------------------

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание Au и Au+1 выражено не в двоично-десятичном коде. Переводится в состояние ON, когда содержание Ad и Ad+1 выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.

Наименование	Обозначение	Функционирование
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды осуществляется перенос. OFF в других случаях.

**Меры предосторожности**

Если содержание слов Au, Au+1 и Ad, Ad+1 выражено не в двоично-десятичном коде, определяется ошибка и флаг ошибки переводится в состояние ON.

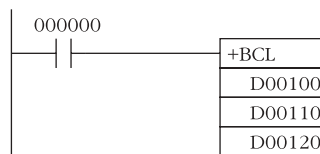
Когда в результате выполнения команды содержание R и R+1 равно 00000000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате сложения выполняется перенос, флаг переноса переводится в состояние ON.

**Примечание:** Для сброса Флага переноса (CY) выполняйте команду (CLC041).

**Примеры**

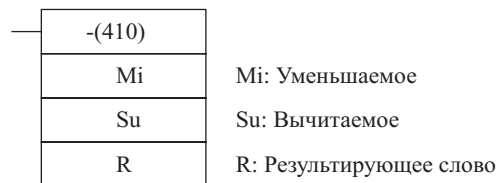
В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится суммирование содержания слов D00101, D00100 и D00111, D00110 в виде восьмизначных двоично-десятичных чисел, и флага переноса. Результат записывается в D00121 и D00120.



### 3-11-9 Команда вычитания двоичных данных со знаком без переноса SIGNED BINARY SUBSTRACT WITHOUT CARRY: - (410)

**Назначение**

По команде - (410) осуществляется вычитание четырехзначных шестнадцатеричных данных (простого слова) и/или констант.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	- (410)
	Выполняется один раз для дифференцирования вверх	@ - (410)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Mi	Su	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D0000...D32767		
Область EM, не содержащая банков	E00000...E32767		

Область	Mi	Su	R
Область EM, содержащая банки	En_0000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF (двоичные)	-	
Регистры данных	DR0...DR15		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15		

**Описание**

По команде -(410) осуществляется вычитание двоичного значения числа, находящегося в Su, из двоичного значения числа, находящегося в Mi, и вывод результата в слово R.

Когда результат является отрицательным числом, он выводится в слово R в виде дополнения по модулю 2. (Для ознакомления с примером вычисления дополнения по модулю 2 обратитесь к разделу 3-11-10 «DOUBLE SIGNED BINARY SUBSTRACT WITHOUT CARRY: - L(411)»).

(Двоичное со знаком)

-  (Двоичное со знаком)

Флаг переноса CY переводится в состояние ON, когда в результате вычитания выполняется отрицательный перенос

(Двоичное со знаком)

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды осуществляется отрицательный перенос. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, когда результат вычитания отрицательного числа из положительного числа находится в пределах от 8000 до FFFF (шестн.). OFF в других случаях.
Флаг отрицательного переполнения	UF	Переводится в состояние ON, когда результат вычитания положительного числа из отрицательного числа находится в пределах от 0000 до 7FFF (шестн.). OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда в результате вычитания бит старшего разряда результирующего слова равен 1. OFF в других случаях.

**Меры предосторожности**

После выполнения команды -(410) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения команды результирующее слово R равно 0000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате вычитания выполняется отрицательный перенос, флаг переноса переводится в состояние ON.

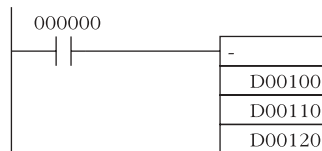
Когда результат вычитания отрицательного числа из положительного числа является отрицательным числом (в пределах от 8000 до FFFF шестн.), флаг переполнения переводится в состояние ON.

Когда результат вычитания положительного числа из отрицательного числа является положительным числом (в пределах от 0000 до 7FFF шестн.) флаг отрицательного переполнения переводится в состояние ON.

Если в результате выполнения команды содержание старшего бита слова R равно 1, флаг отрицательного значения переводится в состояние ON.

### Примеры

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится вычитание содержания слова D00100 из содержания слова D00110, в виде двоичных значений четырехзначных чисел со знаком. Результат записывается в D00120.

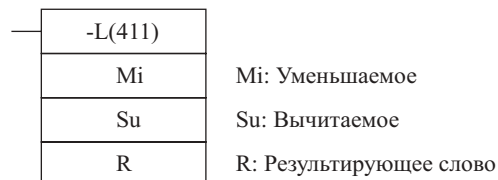


### 3-11-10 Команда вычитания двойных двоичных данных со знаком без переноса DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY: -L(411)

#### Назначение

По команде -L(411) осуществляется вычитание восьмизначных шестнадцатеричных данных (двойного слова) и/или констант.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	-L(411)
	Выполняется один раз для дифференцирования вверх	@ -L(411)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Спецификации операндов

Область	Mi	Su	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142		
Рабочая область	W000...W510		
Область удержания бита (Holding bit area)	H000...H510		
Область вспомогательных битов	A000...A958	A448...A958	
Область таймера	T0000...T4094		
Область счетчика	C0000...C4094		
Область DM	D00000...D32766		
Область EM, не содержащая банков	E00000...E32766		
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#00000000...#FFFF FFFF (двоичные)	-	
Регистры данных	-		
Индексные регистры	IR0...IR15		

Область	Mi	Su	R
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15		

**Описание**

По команде -L(411) осуществляется вычитание двоичных значений чисел, находящихся в Su и Su+1, из двоичных значений чисел, находящихся в Mi и Mi+1. Результат выводится в слова R и R+1.

Mi+1    Mi    (Двоичные со знаком)

-    Su+1    Su    (Двоичные со знаком)

Флаг переноса CY переводится в состояние ON, когда в результате вычитания выполняется отрицательный перенос

CY    R+1    R    (Двоичные со знаком)

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды осуществляется отрицательный перенос. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, когда результат вычитания отрицательного числа из положительного числа находится в пределах от 80000000 до FFFFFFFF (шестн.). OFF в других случаях.
Флаг отрицательного переполнения	UF	Переводится в состояние ON, когда результат вычитания положительного числа из отрицательного числа находится в пределах от 00000000 до 7FFFFFFF (шестн.). OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда в результате вычитания бит старшего разряда результирующего слова равен 1. OFF в других случаях.

**Меры предосторожности**

Перед выполнением команды -L(411) непременно выполняйте команду очистки флага переноса (CLC(041)).

После выполнения команды -L(411) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения команды содержание результирующих слов R и R+1 равно 00000000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате вычитания выполняется отрицательный перенос, флаг переноса переводится в состояние ON.

Когда результат вычитания отрицательного числа из положительного числа является отрицательным числом (в пределах от 80000000 до FFFFFFFF шестн.), флаг переполнения переводится в состояние ON.

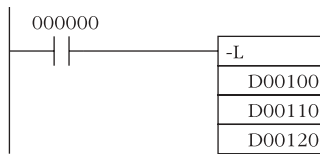
Когда результат вычитания положительного числа из отрицательного числа является положительным числом (в пределах от 00000000 до 7FFFFFFF шестн.) флаг отрицательного переполнения переводится в состояние ON.

Если в результате выполнения команды содержание старшего бита слова R+1 равно 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, содержание слов D00111, D00110 вычитается из содержания слов D00101, D00100 в виде двоичных значений восьмизначных чисел со знаком. Результат записывается в D00121 и D00120.





Если результат является отрицательным числом ( $M_i < S_u$  или  $M_i + 1, M_i < S_u + 1, S_u$ ), данные выводятся в виде дополнения по модулю 2, и флаг переноса переводится в состояние ON, для индикации отрицательного значения результата. Для преобразования дополнения по модулю 2 в истинное значение необходима команда, осуществляющая вычитание результата из 0, применяя в этом случае флаг переноса (CY) в качестве условия выполнения.

**Примечание: дополнение по модулю 2**

Дополнение по модулю 2 – это значение, получаемое посредством вычитания каждой двоичной цифры из единицы, и прибавления единицы к результату. Например, дополнение по модулю 2 к числу 1101 вычисляется следующим образом: 1111 (F шестн.) - 1101 (D шестн.) + 1 (1 шестн.) = 0011 (3 шестн.). Дополнение по модулю 2 к числу 3039 (шестн.) вычисляется следующим образом: FFFF (шестн.) - 3039 (шестн.) + 0001 (шестн.) = CFC7. Таким образом, дополнение четырехзначного шестнадцатеричного числа по модулю 2 вычисляется по правилу: FFFF (шестн.) - a (шестн.) + 0001 (шестн.) = b (шестн.). Для вычисления истинного значения из дополнения выполняются следующие действия: a (шестн.) = 10000 (шестн.) - b (шестн.). Например, истинным значением числа, дополнение к которому равно CFC7, является: 10000 (шестн.) - CFC7 = 3039

**Пример 1**

	Данные со знаком	Данные без знака
$\begin{array}{r} \text{F F F F} \\ - \text{0 0 0 0} \\ \hline \text{F F F E} \end{array}$	$\begin{array}{r} -1 \\ -+1 \\ \hline -2 \\ \text{(См. прим. 1)} \end{array}$	$\begin{array}{r} 6\ 5\ 5\ 3\ 5 \\ - \phantom{0000} \\ \hline 6\ 5\ 5\ 3\ 4 \\ \text{(См. прим. 2)} \end{array}$

Флаг отрицательного значения (N): ON  
Флаг переноса (CY): OFF

- Примечание:**
1. Так как флаг отрицательного значения переводится в состояние ON, результат (FFFF шестн.) является отрицательным числом (дополнение по модулю 2) и, следовательно, равен -2.
  2. Так как флаг переноса переводится в состояние OFF, результат (FFFE шестн.) является положительным числом без знака и равен 65534.

**Пример 2**

	Данные со знаком	Данные без знака
$\begin{array}{r} \text{F F F D} \\ - \text{F F F F} \\ \hline \text{F F F E} \end{array}$	$\begin{array}{r} -3 \\ -1 \\ \hline -2 \\ \text{(См. прим. 1)} \end{array}$	$\begin{array}{r} 6\ 5\ 5\ 3\ 3 \\ - 6\ 5\ 5\ 3\ 5 \\ \hline 6\ 5\ 5\ 3\ 4 \\ \text{(См. прим. 2)} \end{array}$

Флаг отрицательного значения (N): ON  
Флаг переноса (CY): OFF

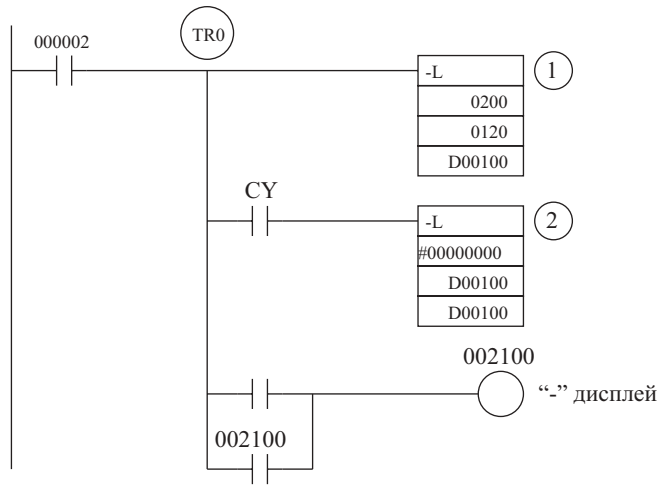
- Примечание:**
1. Так как флаг отрицательного значения переводится в состояние ON, результат (FFFE шестн.) является отрицательным числом (дополнение по модулю 2) и, следовательно, равен -2.
  2. Так как флаг переноса переводится в состояние ON, результат (FFFE шестн.) является отрицательным числом (дополнение по модулю 2) и, следовательно, равен -2 после преобразования в истинное значение.

**Пример программы**

20F55A10 - B8A360E3 = -97AE06D3

В данном примере двоичное значение восьмизначного числа в СЮ 0121 и СЮ 0120 вычитается из двоичного значения числа в D00101 и D 00100, а результат выводится в виде восьмизначного числа в D 00101 и

D 00100. В случае, когда результат является отрицательной величиной, выполняется команда в ветви (2), после чего истинное значение выводится в D00101 и D00100.



**Вычитание в ветви 1**

Mi+1: CIO 0201      Mi: CIO 0200

2	0	F	5	5	A	1	0
---	---	---	---	---	---	---	---

Su+1: CIO 0121      Su: CIO 01200

B	8	A	3	6	0	E	3
---	---	---	---	---	---	---	---

CY	R+1: D00101	R: D00100
1	6 8 A 1	F 9 2 D

Флаг переноса находится в состоянии ON, поэтому результат вычитается из 00000000 для вычисления истинного значения.

**Вычитание в ветви 2**

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

Su+1: D00101      Su: D00100

6	8	5	1	F	9	2	D
---	---	---	---	---	---	---	---

CY	R+1: D00101	R: D00100
1	9 7 A E	0 6 D 3

**Конечный результат вычитания**

Mi+1: CIO 0201      Mi: CIO 0200

2	0	F	5	5	A	1	0
---	---	---	---	---	---	---	---

Su+1: CIO 0121      Su: CIO 01200

6	8	5	1	F	9	2	D
---	---	---	---	---	---	---	---

CY	R+1: D00101	R: D00100
1	9 7 A E	0 6 D 3

Флаг переноса (CY) переводится в состояние ON, поэтому истинным значением является -97AE06D3. Так как содержанием D00101 и D00100 является отрицательное число, флаг переноса используется для перевода в состояние ON удерживающего бита, который в свою очередь переводит в состояние ON бит, индицирующий отрицательное значение.

**3-11-11 Команда вычитания двоичных данных со знаком с переносом SIGNED BINARY SUBSTRACT WITH CARRY: - C(412)**

**Назначение**

По команде - C(412) осуществляется вычитание четырехзначных шестнадцатеричных данных (простого слова) и/или констант с использованием флага переноса (CY).

**Символ релейно-контактной схемы**

-C(412)	
Mi	Mi: Уменьшаемое
Su	Su: Вычитаемое
R	R: Результирующее слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	- C(412)
	Выполняется один раз для дифференцирования вверх	@ - C(412)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Mi	Su	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF (двоичные)	-	
Регистры данных	DR0...DR15		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(-)IR0...,(-)IR15		

**Описание**

По команде -C(412) осуществляется вычитание двоичного значения числа, находящегося в Su, и флага переноса из двоичного значения числа, находящегося в Mi. Результат выводится в слово R. Когда результат является отрицательным числом, он записывается в R в виде дополнения по модулю 2.

(Двоичное со знаком)

(Двоичное со знаком)

-

Флаг переноса CY переводится в состояние ON, когда в результате вычитания выполняется отрицательный перенос

(Двоичное со знаком)

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды осуществляется отрицательный перенос. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, когда результат вычитания отрицательного числа и флага переноса из положительного числа находится в пределах от 8000 до FFFF (шестн.). OFF в других случаях.
Флаг отрицательного переполнения	UF	Переводится в состояние ON, когда результат вычитания положительного числа и флага переноса из отрицательного числа находится в пределах от 0000 до 7FFF (шестн.). OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда в результате вычитания бит старшего разряда результирующего слова равен 1. OFF в других случаях.

**Меры предосторожности**

После выполнения команды -C(412) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения команды результирующее слово R равно 0000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате вычитания выполняется отрицательный перенос, флаг переноса переводится в состояние ON.

Когда результат вычитания отрицательного числа и флага переноса из положительного числа является отрицательным числом (в пределах от 8000 до FFFF шестн.), флаг переполнения переводится в состояние ON.

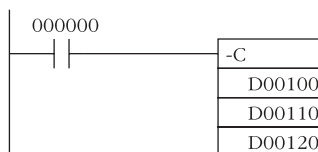
Когда результат вычитания положительного числа и флага переноса из отрицательного числа является положительным числом (в пределах от 0000 до 7FFF шестн.) флаг отрицательного переполнения переводится в состояние ON.

Если в результате выполнения команды содержание старшего бита слова R равно 1, флаг отрицательного значения переводится в состояние ON.

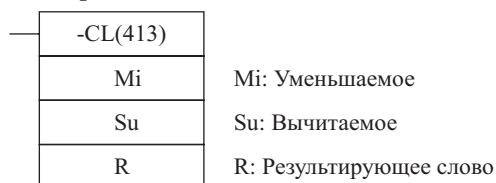
**Примечание:** Для сброса Флага переноса (CY) выполняйте команду (CLC041).

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится вычитание содержания слова D00110 и флага переноса из содержания слова D00100, в виде двоичных значений четырехзначных чисел со знаком. Результат записывается в D00120.

**3-11-12 Команда вычитания с переносом двойных двоичных данных со знаком DOUBLE SIGNED BINARY SUBSTRACT WITH CARRY: -CL(413)****Назначение**

По команде -CL(413) осуществляется вычитание восьмизначных шестнадцатеричных данных (двойного слова) и/или констант с использованием флага переноса (CY).

**Символ релейно-контактной схемы**

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	-CL(413)
	Выполняется один раз для дифференцирования вверх	@ -CL(413)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Mi	Su	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142		
Рабочая область	W000...W510		
Область удержания бита (Holding bit area)	H000...H510		
Область вспомогательных битов	A000...A958	A448...A958	
Область таймера	T0000...T4094		
Область счетчика	C0000...C4094		
Область DM	D00000...D32766		
Область EM, не содержащая банков	E00000...E32766		
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#00000000...#FFFF FFFF (двоичные)	-	
Регистры данных	-		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15		

**Описание**

По команде -CL(413) осуществляется вычитание двоичных значений чисел, находящихся в Su и Su+1, и знака переноса из двоичных значений чисел, находящихся в Mi и Mi+1. Результат выводится в слова R и R+1. Когда результат является отрицательной величиной, он выводится в слова R и R+1 в виде дополнения по модулю 2.

Mi+1	Mi	(Двоичные со знаком)
------	----	----------------------

Su+1	Su	(Двоичные со знаком)
------	----	----------------------

-

CY
----

Флаг переноса CY переводится в состояние ON, когда в результате вычитания выполняется отрицательный перенос

CY	R+1	R	(Двоичные со знаком)
----	-----	---	----------------------

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.

Наименование	Обозначение	Функционирование
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды осуществляется отрицательный перенос. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, когда результат вычитания отрицательного числа и флага переноса из положительного числа находится в пределах от 80000000 до FFFFFFFF (шестн.). OFF в других случаях.
Флаг отрицательного переполнения	UF	Переводится в состояние ON, когда результат вычитания положительного числа и флага переноса из отрицательного числа находится в пределах от 00000000 до 7FFFFFFF (шестн.). OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда в результате вычитания бит старшего разряда результирующего слова равен 1. OFF в других случаях.

### Меры предосторожности

После выполнения команды -CL(413) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения команды содержание результирующих слов R и R+1 равно 00000000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате вычитания выполняется отрицательный перенос, флаг переноса переводится в состояние ON.

Когда результат вычитания отрицательного числа и флага переноса из положительного числа является отрицательным числом (в пределах от 80000000 до FFFFFFFF шестн.), флаг переполнения переводится в состояние ON.

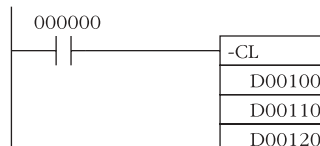
Когда результат вычитания положительного числа и флага переноса из отрицательного числа является положительным числом (в пределах от 00000000 до 7FFFFFFF шестн.) флаг отрицательного переполнения переводится в состояние ON.

Если в результате выполнения команды содержание старшего бита слова R+1 равно 1, флаг отрицательного значения переводится в состояние ON.

**Примечание:** Для сброса Флага переноса (CY) выполняйте команду (CLC041).

### Примеры

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, содержание слов D00111, D00110 и флаг переноса вычитаются из содержания слов D00101, D00100 в виде двоичных значений восьмизначных чисел со знаком. Результат записывается в D00121 и D00120.



Если результат выполнения вычитания является отрицательным числом ( $M_i < S_u$  или  $M_{i+1}, M_i < S_{u+1}, S_u$ ), данные выводятся в виде дополнения по модулю 2. Флаг переноса переводится в состояние ON. Для преобразования дополнения по модулю 2 в истинное значение необходима программа, осуществляющая вычитание результата из 0, применяя в этом случае флаг переноса (CY) в качестве условия выполнения. Таким образом, переключение флага переноса в состояние ON указывает на отрицательное значение результата.

#### Примечание: дополнение по модулю 2

Дополнение по модулю 2 – это значение, получаемое посредством вычитания каждой двоичной цифры из единицы, и прибавления единицы к результату.

Пример: Дополнение по модулю 2 к числу 1101 вычисляется следующим образом: 1111 (F шестн.) - 1101 (D шестн.) + 1 (1 шестн.) = 0011 (3 шестн.).

Пример: Дополнение по модулю 2 к числу 3039 (шестн.) вычисляется следующим образом: FFFF (шестн.) - 3039 (шестн.) + 0001 (шестн.) = CFC7.

Таким образом, дополнение четырехзначного шестнадцатеричного числа по модулю 2 вычисляется по правилу:

FFFF (шестн.) - a (шестн.) + 0001 (шестн.) = b (шестн.).

Для вычисления истинного значения «a»(шестн.) из дополнения «b» (шестн.) по модулю 2, выполняются следующие действия:

a (шестн.) = 10000 (шестн.) - b (шестн.).

Пример: Истинным значением числа, дополнение к которому равно CFC7, является:

10000 (шестн.) - CFC7 (шестн.) = 3039(шестн.)

## 3-11-13 Команда вычитания двоично-десятичных данных без переноса BCD SUBTRACT WITHOUT CARRY: -B(414)

**Назначение**

По команде -B(414) осуществляется вычитание четырехзначных двоично-десятичных данных (простого слова) и/или констант.

**Символ релейно-контактной схемы**

-B(414)	
Mi	Mi: Уменьшаемое
Su	Su: Вычитаемое
R	R: Результирующее слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	-B(414)
	Выполняется один раз для дифференцирования вверх	@ -B(414)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

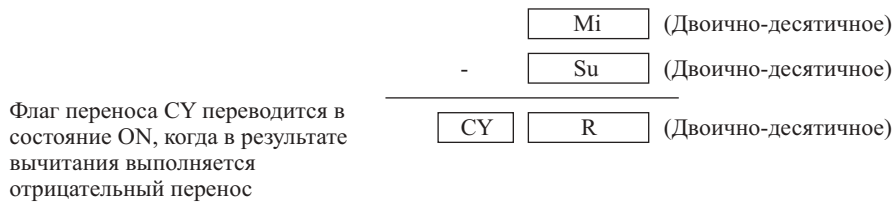
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Mi	Su	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	0000...9999 (двоично-десятичные)	-	
Регистры данных	DR0...DR15		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15		

**Описание**

По команде -B(414) осуществляется вычитание двоично-десятичного числа, находящегося в Su, из двоично-десятичного числа, находящегося в Mi, и вывод результата в слово R. Если результат вычитания является отрицательным числом, вывод результата осуществляется в виде дополнения по модулю 10.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание Mi выражено не в двоично-десятичном коде. Переводится в состояние ON, когда содержание Su выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды осуществляется отрицательный перенос. OFF в других случаях.

**Меры предосторожности**

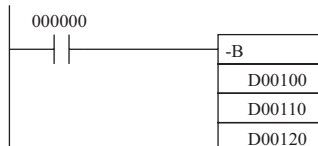
В случае, когда Mi и/или Su выражены не в двоично-десятичном коде, определяется ошибка и флаг ошибки переводится в состояние ON.

Когда в результате выполнения команды содержание результирующего слова R равно 0000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате выполнения вычитания осуществляется отрицательный перенос, флаг переноса переводится в состояние ON.

**Примеры**

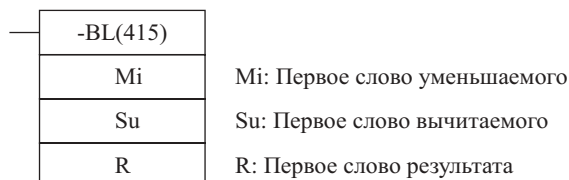
В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится вычитание содержания слова D00100 из содержания слова D00110, в виде четырехзначных двоично-десятичных чисел. Результат записывается в D00120.



### 3-11-14 Команда вычитания двойных двоично-десятичных данных без переноса DOUBLE BCD SUBSTRACT WITHOUT CARRY: -BL(415)

**Назначение**

По команде -BL(415) осуществляется вычитание восьмизначных двоично-десятичных данных (двойного слова) и/или констант.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	-BL(415)
	Выполняется один раз для дифференцирования вверх	@ -BL(415)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	



**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Mi	Su	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142		
Рабочая область	W000...W510		
Область удержания бита (Holding bit area)	H000...H510		
Область вспомогательных битов	A000...A958	A448...A958	
Область таймера	T0000...T4094		
Область счетчика	C0000...C4094		
Область DM	D00000...D32766		
Область EM, не содержащая банков	E00000...E32766		
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#00000000...#99999999 999 (двоично-десятичные)	-	
Регистры данных	-		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(++) ,(-)IR0...,-(-)IR15		

**Описание**

По команде -BL(415) осуществляется вычитание двоично-десятичных чисел, находящихся в Su и Su+1, из двоично-десятичных чисел, находящихся в Mi и Mi+1. Результат выводится в слова R и R+1. Если результат является отрицательным числом, вывод результата осуществляется в виде дополнения по модулю 10.

$$\begin{array}{r}
 \boxed{Mi+1} \quad \boxed{Mi} \quad (\text{Двоично-десятичные}) \\
 - \quad \boxed{Su+1} \quad \boxed{Su} \quad (\text{Двоично-десятичные})
 \end{array}$$

Флаг переноса CY переводится в состояние ON, когда в результате вычитания выполняется отрицательный перенос

$$\boxed{CY} \quad \boxed{R+1} \quad \boxed{R} \quad (\text{Двоично-десятичные})$$

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание Mi и/или M+1 выражено не в двоично-десятичном коде. Переводится в состояние ON, когда содержание Su и/или Su+1 выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды осуществляется отрицательный перенос. OFF в других случаях.

**Меры предосторожности**

В случае, когда Mi, M+1 и/или Su, Su+1 выражены не в двоично-десятичном коде, определяется ошибка и флаг ошибки переводится в состояние ON.

Когда в результате выполнения команды содержание результирующих слов R, R+1 равно 00000000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате выполнения вычитания осуществляется отрицательный перенос, флаг переноса переводится в состояние ON.

### Примеры

В следующем ниже примере, когда СЮ 000000 находится в состоянии ON, содержание слов D00111, D00110 вычитается из содержания слов D00101, D00100, в виде восьмизначных двоично-десятичных чисел. Результат записывается в D00121 и D00120.



Если результат является отрицательным числом ( $M_i < S_u$  или  $M_{i+1}, M_i < S_{u+1}, S_u$ ), данные выводятся в виде дополнения по модулю 10. Флаг переноса (CY) переводится в состояние ON. Для преобразования дополнения по модулю 10 в истинное значение необходима программа, осуществляющая вычитание результата из 0, применяя в этом случае флаг переноса (CY) в качестве условия выполнения. Таким образом, переключение флага переноса в состояние ON указывает на отрицательное значение результата.

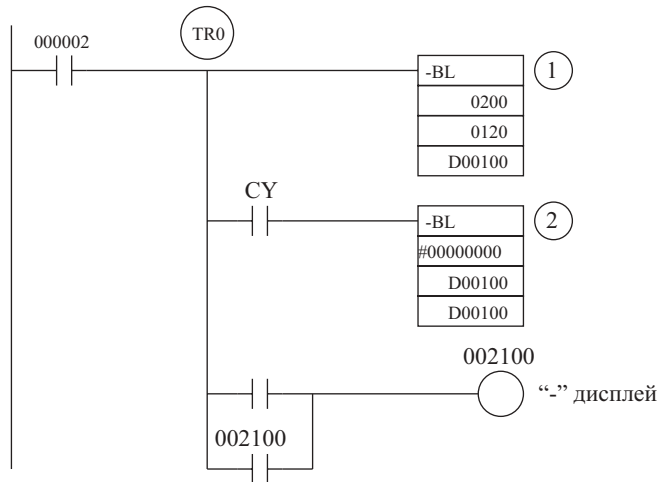
#### Примечание: дополнение по модулю 10

Дополнение по модулю 10 – это значение, получаемое посредством вычитания каждой цифры из 9, и прибавления единицы к результату. Например, дополнение по модулю 10 к числу 7556 вычисляется следующим образом:  $9999 - 7556 + 1 = 2444$ . Таким образом, дополнение A четырехзначного числа по модулю 10 вычисляется по правилу:  $9999 - A + 1 = B$ . Для вычисления истинного значения из дополнения к числу B по модулю 10 выполняется следующее действие:  $A = 10000 - B$ . Например, истинным значением числа, дополнение к которому по модулю 10 равно 2444, является:  $10000 - 2444 = 7556$

### Пример программы

$$9583960 - 17072641 = -7488681$$

В данном примере восьмизначное двоично-десятичное число в СЮ 0121 и СЮ 0120 вычитается из числа в СЮ 0201 и СЮ 0200, а результат выводится в виде восьмизначного двоично-десятичного числа в D 00101 и D 00100. Результат является отрицательной величиной, поэтому выполняется команда в ветви (2), после чего истинное значение выводится в D00101 и D00100.



**Вычитание в ветви 1**

Mi+1: CIO 0201	Mi: CIO 0200
0 9 5 8	3 9 6 0
Su+1: CIO 0121	Su: CIO 01200
1 7 0 7	2 6 4 1

CY	R+1: D00101	R: D00100
1	9 2 5 1	1 3 1 9

09583960+(100000000-17072641)

Флаг переноса находится в состоянии ON, поэтому для вычисления истинного значения результат вычитается из 00000000.

**Вычитание в ветви 2**

0 0 0 0	0 0 0 0
Su+1: D00101	Su: D00100
9 2 5 1	1 3 1 9

CY	R+1: D00101	R: D00100
1	0 7 4 8	8 6 8 1

00000000+(100000000-92511319)

**Конечный результат вычитания**

Mi+1: CIO 0201	Mi: CIO 0200
2 0 F 5	5 A 1 0
Su+1: D00101	Su: D00100
6 8 5 1	F 9 2 D

CY	R+1: D00101	R: D00100
1	0 7 4 8	8 6 8 1

Флаг переноса (CY) переводится в состояние ON, поэтому истинным значением является -7488681. Так как содержанием D00101 и D00100 является отрицательное число, флаг переноса используется для перевода в состояние ON удерживающего бита, который в свою очередь переводит в состояние ON бит, индицирующий отрицательное значение.

### 3-11-15 Команда вычитания двоично-десятичных данных с переносом BCD SUBSTRACT WITH CARRY: -BC(416)

**Назначение**

По команде - BC(416) осуществляется вычитание четырехзначных двоично-десятичных данных (простого слова) и/или констант с использованием флага переноса.

**Символ релейно-контактной схемы**

-BC(416)	
Mi	Mi: Уменьшаемое
Su	Su: Вычитаемое
R	R: Результирующее слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	- BC(416)
	Выполняется один раз для дифференцирования вверх	@ - BC(416)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Mi	Su	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#9999 (двоично-десятичные)	-	
Регистры данных	DR0...DR15		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15		

**Описание**

По команде -BC(416) производится вычитание двоично-десятичного числа, находящегося в Su, и флага переноса из двоично-десятичного числа, находящегося в Mi. Результат выводится в слово R. Если результат является отрицательным числом, вывод результата осуществляется в виде дополнения по модулю 10.

(Двоично-десятичное)

(Двоично-десятичное)

-

Флаг переноса CY переводится в состояние ON, когда в результате вычитания выполняется отрицательный перенос

(Двоично-десятичное)

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание Mi выражено не в двоично-десятичном коде. Переводится в состояние ON, когда содержание Su выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды осуществляется отрицательный перенос. OFF в других случаях.

**Меры предосторожности**

В случае, когда Mi и/или Su выражены не в двоично-десятичном коде, определяется ошибка и флаг ошибки переводится в состояние ON.

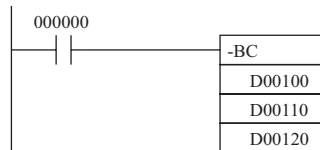
Когда в результате выполнения команды содержание результирующего слова R равно 0000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате выполнения вычитания осуществляется отрицательный перенос, флаг переноса переводится в состояние ON.

**Примечание:** Для сброса Флага переноса (CY) выполняйте команду (CLC041).

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится вычитание содержания слова D00100 и флага переноса из содержания слова D00110, в виде четырехзначных двоично-десятичных чисел. Результат записывается в D00120.



### 3-11-16 Команда вычитания с переносом двойных двоично-десятичных данных DOUBLE BCD SUBSTRACT WITH CARRY: -BCL(417)

**Назначение**

По команде -BCL(417) осуществляется вычитание восьмизначных двоично-десятичных данных (двойного слова) и/или констант с использованием флага переноса.

**Символ релейно-контактной схемы**

-BCL(417)	
Mi	Mi: Первое слово уменьшаемого
Su	Su: Первое слово вычитаемого
R	R: Первое слово результата

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	-BCL(417)
	Выполняется один раз для дифференцирования вверх	@ -BCL(417)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

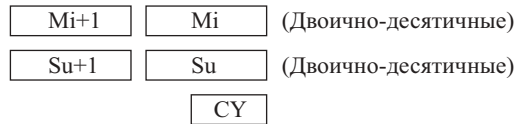
**Спецификации операндов**

Область	Mi	Su	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142		
Рабочая область	W000...W510		
Область удержания бита (Holding bit area)	H000...H510		
Область вспомогательных битов	A000...A958	A448...A958	
Область таймера	T0000...T4094		
Область счетчика	C0000...C4094		
Область DM	D00000...D32766		
Область EM, не содержащая банков	E00000...E32766		
Область EM, содержащая банки	En_ 00000...En_ 32766 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		

Область	Mi	Su	R
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#00000000...#99999999 (двоично-десятичные)	–	
Регистры данных	–		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...IR15(++) ,(- -)IR0...(- -)IR15		

**Описание**

По команде -BCL(417) осуществляется вычитание двоично-десятичных чисел, находящихся в Su и Su+1, и флага переноса из двоично-десятичных чисел, находящихся в Mi и Mi+1. Результат выводится в слова R и R+1. Если результат является отрицательным числом, вывод результата осуществляется в виде дополнения по модулю 10.



Флаг переноса CY переводится в состояние ON, когда в результате вычитания выполняется отрицательный перенос



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание Mi и/или Mi+1 выражено не в двоично-десятичном коде. Переводится в состояние ON, когда содержание Su и/или Su+1 выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда при выполнении команды осуществляется отрицательный перенос. OFF в других случаях.

**Меры предосторожности**

В случае, когда Mi, Mi+1 и/или Su, Su+1 выражены не в двоично-десятичном коде, определяется ошибка и флаг ошибки переводится в состояние ON.

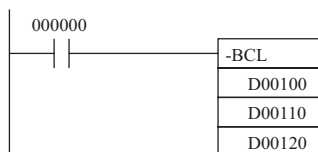
Когда в результате выполнения команды содержание результирующих слов R, R+1 равно 00000000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате выполнения вычитания осуществляется отрицательный перенос, флаг переноса переводится в состояние ON.

**Примечание:** Для сброса Флага переноса (CY) выполняйте команду (CLC041).

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, содержание слов D00111, D00110 и флаг переноса вычитается из содержания слов D00101, D00100, в виде восьмизначных двоично-десятичных чисел. Результат записывается в D00121 и D00120.



Если результат является отрицательным числом ( $M_i < S_u$  или  $M_i + 1, M_i < S_u + 1, S_u$ ), данные выводятся в виде дополнения по модулю 10. Флаг переноса (CY) переводится в состояние ON. Для преобразования дополнения по модулю 10 в истинное значение необходима программа, осуществляющая вычитание результата из 0, применяя в этом случае флаг переноса (CY) в качестве условия выполнения. Таким образом, переключение флага переноса в состояние ON указывает на отрицательное значение результата.

**Примечание: дополнение по модулю 10**

Дополнение по модулю 10 – это значение, получаемое посредством вычитания каждой цифры из 9, и прибавления единицы к результату. Например, дополнение по модулю 10 к числу 7556 вычисляется следующим образом:  $9999 - 7556 + 1 = 2444$ . Таким образом, дополнение по модулю 10 четырехзначного числа A вычисляется по правилу:  $9999 - A + 1 = B$ . Для вычисления истинного значения A из дополнения по модулю 10 выполняется следующее действие:  $A = 10000 - B$ . Например, истинным значением числа, дополнение к которому по модулю 10 является 2444, является:  $10000 - 2444 = 7556$ .

### 3-11-17 Команда перемножения двоичных данных со знаком SIGNED BINARY MULTIPLY: \*(420)

**Назначение**

По команде \*(420) осуществляется перемножение четырехзначных шестнадцатеричных данных и/или констант.

**Символ релейно-контактной схемы**

*(420)	
Md	Md: Множимое
Mr	Mr: Множитель
R	R: Результирующее слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	*(420)
	Выполняется один раз для дифференцирования вверх	@ *(420)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

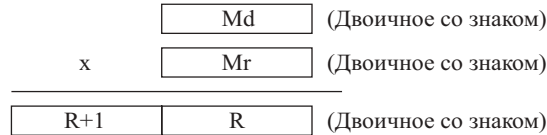
**Спецификации операндов**

Область	Md	Mr	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6142	
Рабочая область	W000...W511	W000...W510	
Область удержания бита (Holding bit area)	H000...H511	H000...H510	
Область вспомогательных битов	A000...A959	A448...A958	
Область таймера	T0000...T4095	T0000...T4094	
Область счетчика	C0000...C4095	C0000...C4094	
Область DM	D00000...D32767	D00000...D32766	
Область EM, не содержащая банков	E00000...E32767	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF (двоичные)	–	
Регистры данных	DR0...DR15	–	
Индексные регистры	–		

Область	Md	Mr	R
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(++) ,-( -)IR0...,-( -)IR15		

**Описание**

По команде \*(420) осуществляется перемножение двоичных значений чисел, находящихся в Md и Mr. Результат выводится в слова R и R+1.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующего слова равен 1. OFF в других случаях.

**Меры предосторожности**

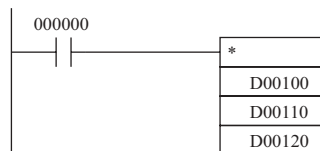
После выполнения команды \*(420) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения умножения содержание результирующего слова R равно 0000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате выполнения команды старший бит результирующих слов R+1 и R равен 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

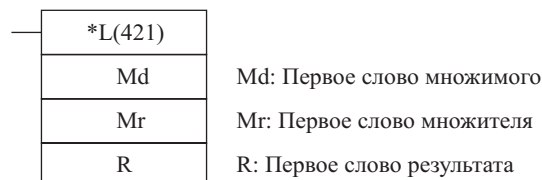
В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится перемножение содержания слов D00100 и D00110, в виде двоичных значений четырехзначных чисел. Результат записывается в D00120.



### 3-11-18 Команда перемножения двойных двоичных данных со знаком DOUBLE SIGNED BINARY MULTIPLY: \*L(421)

**Назначение**

По команде \*L(421) осуществляется перемножение восьмизначных шестнадцатеричных данных и/или констант.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	*L(421)
	Выполняется один раз для дифференцирования вверх	@ *L(421)
	Выполняется один раз для дифференцирования вниз	Не поддерживается



Указание немедленной ре-генерации	Не поддерживается
-----------------------------------	-------------------

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Md	Mr	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	CIO 0000...CIO 6140	
Рабочая область	W000...W510	W000...W508	
Область удержания бита (Holding bit area)	H000...H510	H000...H508	
Область вспомогательных битов	A000...A958	A448...A956	
Область таймера	T0000...T4094	T0000...T4092	
Область счетчика	C0000...C4094	C0000...C4092	
Область DM	D00000...D32766	D00000...D32764	
Область EM, не содержащая банков	E00000...E32766	E00000...E32764	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	En_00000...En_32764 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#00000000...#FFFF FFFF (двоичные)	–	
Регистры данных	–		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047, IR0 -2048...+2047, IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(++) ,(-)IR0...,-(-)IR15		

**Описание**

По команде \*L(421) осуществляется перемножение двоичных значений чисел со знаком, находящихся в Md, Md+1 и Mr, Mr+1. Результат выводится в слова R, R+1, R+2 и R+3.

	Md+1	Md	(Двоичное со знаком)		
x	Mr+1	Mr	(Двоичное со знаком)		
	R+3	R+2	R+1	R	(Двоичное со знаком)

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующего слова равен 1. OFF в других случаях.

**Меры предосторожности**

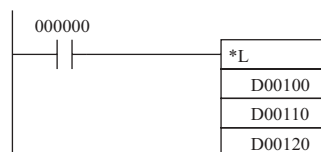
После выполнения команды \*L(421) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения умножения содержание результирующих слов R, R+1, R+2, R+3 равно 0000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате выполнения команды старший бит результирующего слова R+1 равен 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

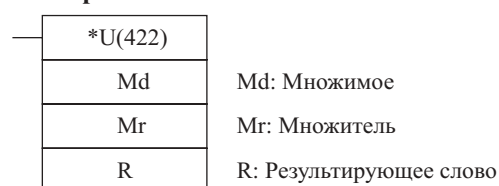
В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится перемножение содержания слов D00100, D00110, D00111 и D00112, в виде двоичных значений восьмизначных чисел. Результат записывается в D00121 и D00120.



### 3-11-19 Команда перемножения двоичных данных без знака UNSIGNED BINARY MULTIPLY: \*U(422)

**Назначение**

По команде \*U(422) осуществляется перемножение четырехзначных шестнадцатеричных данных и/или констант.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	*U(422)
	Выполняется один раз для дифференцирования вверх	@ *U(422)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

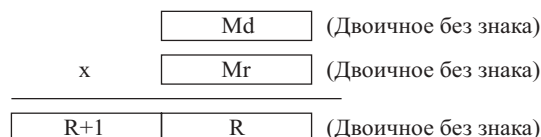
**Спецификации операндов**

Область	Md	Mг	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6142	
Рабочая область	W000...W511	W000...W510	
Область удержания бита (Holding bit area)	H000...H511	H000...H510	
Область вспомогательных битов	A000...A959	A448...A958	
Область таймера	T0000...T4095	T0000...T4094	
Область счетчика	C0000...C4095	C0000...C4094	
Область DM	D00000...D32767	D00000...D32766	
Область EM, не содержащая банков	E00000...E32767	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF (двоичные)	–	
Регистры данных	DR0...DR15	–	
Индексные регистры	–		

Область	Md	Mr	R
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,(- -)IR0...(- -)IR15		

**Описание**

По команде \*U(422) осуществляется перемножение двоичных значений чисел, находящихся в Md и Mr. Результат выводится в слова R и R+1.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующего слова равен 1. OFF в других случаях.

**Меры предосторожности**

После выполнения команды \*U(422) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения умножения содержание результирующих слов R и R+1 равно 0000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате выполнения команды старший бит результирующего слова R+1 равен 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

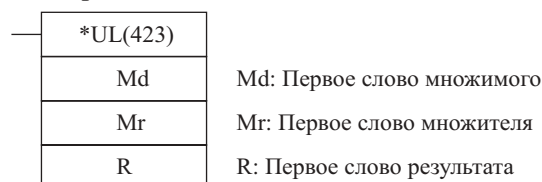
В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится перемножение содержания слов D00100 и D00110, в виде двоичных значений четырехзначных чисел без знака. Результат записывается в D00121 и D00120.



### 3-11-20 Команда перемножения двойных двоичных данных без знака DOUBLE UNSIGNED BINARY MULTIPLY: \*UL(423)

**Назначение**

По команде \*UL(423) осуществляется перемножение восьмизначных шестнадцатеричных данных и/или констант.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	*UL(423)
	Выполняется один раз для дифференцирования вверх	@ *UL(423)
	Выполняется один раз для дифференцирования вниз	Не поддерживается

Указание немедленной ре-генерации	Не поддерживается
-----------------------------------	-------------------

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Md	Mr	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	CIO 0000...CIO 6140	
Рабочая область	W000...W510	W000...W508	
Область удержания бита (Holding bit area)	H000...H510	H000...H508	
Область вспомогательных битов	A000...A958	A448...A956	
Область таймера	T0000...T4094	T0000...T4092	
Область счетчика	C0000...C4094	C0000...C4092	
Область DM	D00000...D32766	D00000...D32764	
Область EM, не содержащая банков	E00000...E32766	E00000...E32764	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	En_00000...En_32764 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#00000000...#FFFF FFFF (двоичные)	–	
Регистры данных	–		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,(- -)IR0...(- -)IR15		

**Описание**

По команде \*UL(423) осуществляется перемножение двоичных значений чисел без знака, находящихся в Md, Md+1 и Mr, Mr+1. Результат выводится в слова R, R+1, R+2 и R+3.

	Md+1	Md	(Двоичные без знака)		
x	Mr+1	Mr	(Двоичные без знака)		
	R+3	R+2	R+1	R	(Двоичные без знака)

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующего слова равен 1. OFF в других случаях.

**Меры предосторожности**

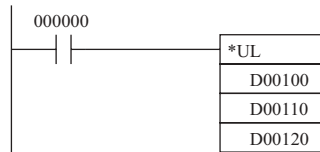
После выполнения команды \*UL(421) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения умножения содержание результирующих слов R, R+1, R+2, R+3 равно 0000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате выполнения команды старший бит результирующих слов R+3 равен 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

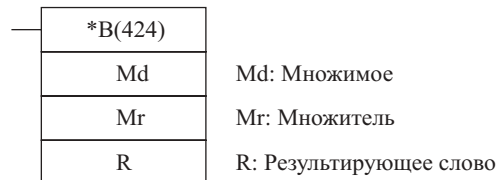
В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится перемножение содержания слов D00101, D00100, D00111 и D00110, в виде двоичных значений восьмизначных чисел без знака. Результат записывается в D00123 D00122 D00121 и D00120.



### 3-11-21 Команда перемножения двоичных данных без знака UNSIGNED BINARY MULTIPLY: \*U(422)

**Назначение**

По команде \*B(424) осуществляется перемножение четырехзначных двоично-десятичных (простого слова) данных и/или констант.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	*B(424)
	Выполняется один раз для дифференцирования вверх	@ *B(424)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

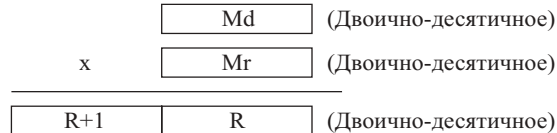
**Спецификации операндов**

Область	Md	Mr	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6142	
Рабочая область	W000...W511	W000...W510	
Область удержания бита (Holding bit area)	H000...H511	H000...H510	
Область вспомогательных битов	A000...A959	A448...A958	
Область таймера	T0000...T4095	T0000...T4094	
Область счетчика	C0000...C4095	C0000...C4094	
Область DM	D00000...D32767	D00000...D32766	
Область EM, не содержащая банков	E00000...E32767	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#9999 (двоично-десятичные)	—	
Регистры данных	DR0...DR15	—	
Индексные регистры	—		

Область	Md	Mг	R
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-(-)IR15		

**Описание**

По команде \*B(424) осуществляется перемножение двоично-десятичных чисел, находящихся в Md и Mг. Результат выводится в слова R и R+1.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание Md выражено не в двоично-десятичном коде. Переводится в состояние ON, когда содержание Mг выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.

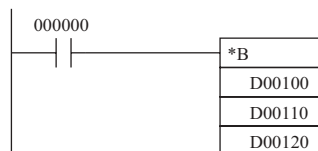
**Меры предосторожности**

В случае, когда Md и/или Mг выражены не в двоично-десятичном коде, определяется ошибка и флаг ошибки переводится в состояние ON.

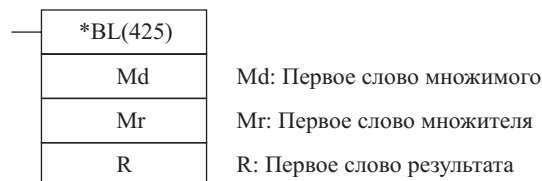
Когда в результате выполнения команды содержание результирующих слов R и R+1 равно 0000 (шестн.), флаг равенства переводится в состояние ON.

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится перемножение содержания слов D00100 и D00110, в виде четырехзначных двоично-десятичных чисел. Результат записывается в D00121 и D00120.

**3-11-22 Команда перемножения двойных двоично-десятичных данных DOUBLE BCD MULTIPLY: \*BL(425)****Назначение**

По команде \*BL(425) осуществляется перемножение восьмизначных двоично-десятичных данных (двойное слово) и/или констант.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	*BL(425)
	Выполняется один раз для дифференцирования вверх	@ *BL(425)
	Выполняется один раз для дифференцирования вниз	Не поддерживается

Указание немедленной ре-генерации	Не поддерживается
-----------------------------------	-------------------

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Md	Mr	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	CIO 0000...CIO 6140	
Рабочая область	W000...W510	W000...W508	
Область удержания бита (Holding bit area)	H000...H510	H000...H508	
Область вспомогательных битов	A000...A958	A448...A956	
Область таймера	T0000...T4094	T0000...T4092	
Область счетчика	C0000...C4094	C0000...C4092	
Область DM	D00000...D32766	D00000...D32764	
Область EM, не содержащая банков	E00000...E32766	E00000...E32764	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	En_00000...En_32764 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	00000000...9999999 9 (двоично-десятичные)	–	
Регистры данных	–		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,-( -)IR15		

**Описание**

По команде \*BL(425) осуществляется перемножение двоично-десятичных чисел, находящихся в Md, Md+1 и Mr, Mr+1. Результат выводится в слова R, R+1, R+2 и R+3.

	Md+1	Md	(Двоично-десятичные)		
x	Mr+1	Mr	(Двоично-десятичные)		
	R+3	R+2	R+1	R	(Двоично-десятичные)

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание Md и/или Md+1 выражено не в двоично-десятичном коде. Переводится в состояние ON, когда содержание Mr и/или Mr+1 выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.

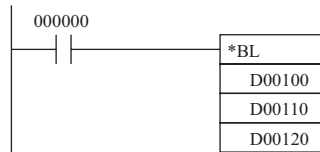
**Меры предосторожности**

В случае, когда Md, Md+1 и/или Mr, Mr+1 выражены не в двоично-десятичном коде, определяется ошибка и флаг ошибки переводится в состояние ON.

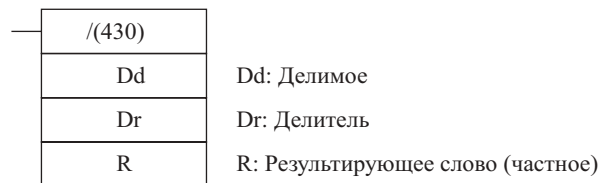
Когда в результате выполнения команды содержание результирующих слов R, R+1, R+2, R+3 равно 00000000 (шестн.), флаг равенства переводится в состояние ON.

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится перемножение содержания слов D00101, D00100, D00111 и D00110, в виде восьмизначных двоично-десятичных чисел. Результат записывается в D00123 D00122 D00121 и D00120.

**3-11-23 Команда деления двоичных данных со знаком SIGNED BINARY DIVIDE: /(430)****Назначение**

По команде /(430) осуществляется деление четырехзначных шестнадцатеричных данных (простого слова) со знаком и/или констант.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	/(430)
	Выполняется один раз для дифференцирования вверх	@/(430)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

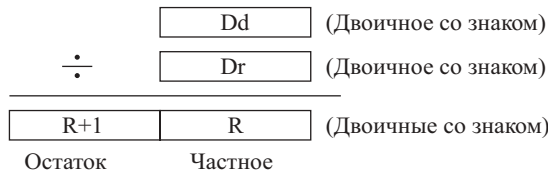
Область	Dd	Dr	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6142	
Рабочая область	W000...W511	W000...W510	
Область удержания бита (Holding bit area)	H000...H511	H000...H510	
Область вспомогательных битов	A000...A959	A448...A958	
Область таймера	T0000...T4095	T0000...T4094	
Область счетчика	C0000...C4095	C0000...C4094	
Область DM	D00000...D32767	D00000...D32766	
Область EM, не содержащая банков	E00000...E32767	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF (двоичные)	#0001...#FFFF (двоичные)	—
Регистры данных	DR0...DR15	—	
Индексные регистры	—		



Область	Dd	Dr	R
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047,IR0...-2048...+2047, IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-(-)IR15		

**Описание**

По команде /L(430) осуществляется деление двоичного значения (16 битов) числа со знаком, находящегося в Dd, на двоичное значение (16 битов) числа со знаком, находящегося в Dr. Результат выводится в слова R и R+1. Частное от деления заносится в R, а остаток - в R+1.



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержание R равно нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующего слова R равен 1. OFF в других случаях.

**Меры предосторожности**

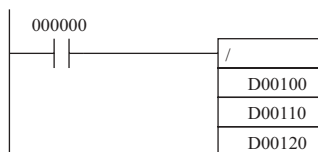
В случае, когда содержание Dr равно нулю, определяется ошибка, и флаг ошибки переводится в состояние ON.

Когда в результате выполнения деления содержание результирующего слова R равно 0000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате выполнения команды старший бит результирующего слова R равен 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится деление содержания слова D00100 на содержание слова D00110, в виде двоичных значений четырехзначных чисел со знаком. Частное от деления записывается в D00120, а остаток – в D00121.

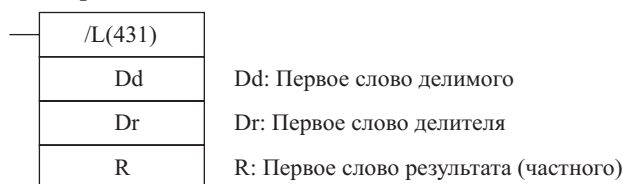


**3-11-24 Команда деления двойных двоичных данных со знаком DOUBLE SIGNED BINARY DIVIDE: /L(431)**

**Назначение**

По команде /L(431) осуществляется деление восьмизначных шестнадцатеричных данных со знаком (двойного слова) и/или констант.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	/L(431)
	Выполняется один раз для дифференцирования вверх	@ /L(431)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Dd	Dr	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	CIO 0000...CIO 6140	
Рабочая область	W000...W510	W000...W508	
Область удержания бита (Holding bit area)	H000...H510	H000...H508	
Область вспомогательных битов	A000...A958	A448...A956	
Область таймера	T0000...T4094	T0000...T4092	
Область счетчика	C0000...C4094	C0000...C4092	
Область DM	D00000...D32766	D00000...D32764	
Область EM, не содержащая банков	E00000...E32766	E00000...E32764	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	En_00000...En_32764 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32766 @E00000...@E32766 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32766 *E00000...*E32766 *En_00000...*En_32766 (n = 0...C)		
Константы	#00000000...#FFFF FFFF (двоичные)	#00000001...#FFFF FFFF (двоичные)	–
Регистры данных	–		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15		

**Описание**

По команде /L(431) осуществляется деление двоичных значений чисел со знаком, находящихся в Dd, Dd+1 и Dr, Dr+1. Результат записывается в слова R, R+1, R+2, R+3. Частное от деления выводится в слова R, R+1, а остаток выводится в слова R+2 и R+3.

÷	Dd+1	Dd	(Двоичные со знаком)
	Dr+1	Dr	(Двоичные со знаком)
R+3	R+2	R+1	R
Остаток		Частное	

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержание R+1, R равно нулю. OFF в других случаях.

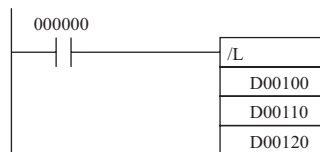
Наименование	Обозначение	Функционирование
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующих слов R+1, R равен 1. OFF в других случаях.

**Меры предосторожности**

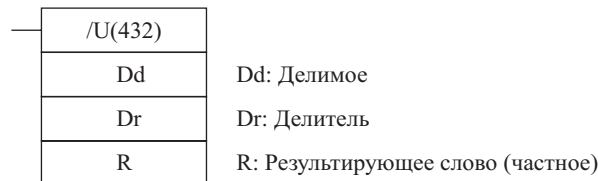
В случае, когда содержание остатка R+3 и R+2 равно нулю, флаг ошибки переводится в состояние ON.  
 Когда в результате выполнения деления содержание результирующих слов R, R+1, равно 00000000 (шестн.), флаг равенства переводится в состояние ON.  
 Если в результате выполнения команды старший бит результирующих слов R+1, R равен 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, содержание слов D00100, D00110, делится на содержание слов D00111 и D00112, в виде двоичных значений восьмизначных чисел со знаком. Частное от деления записывается в D00121 и D00120, а остаток – в D00123 и D 00122..

**3-11-25 Команда деления двоичных данных без знака UNSIGNED BINARY DIVIDE: /U(432)****Назначение**

По команде /U(432) осуществляется деление четырехзначных шестнадцатеричных данных (простого слова) без знака и/или констант.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	/U(432)
	Выполняется один раз для дифференцирования вверх	@ /U(432)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

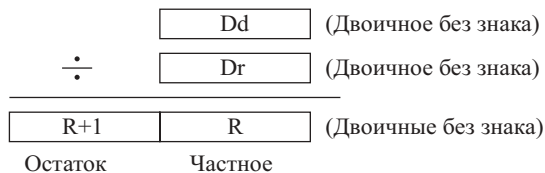
**Спецификации операндов**

Область	Dd	Dr	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6142	
Рабочая область	W000...W511	W000...W510	
Область удержания бита (Holding bit area)	H000...H511	H000...H510	
Область вспомогательных битов	A000...A959	A448...A958	
Область таймера	T0000...T4095	T0000...T4094	
Область счетчика	C0000...C4095	C0000...C4094	
Область DM	D00000...D32767	D00000...D32766	
Область EM, не содержащая банков	E00000...E32767	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32766 (n = 0...C)	

Область	Dd	Dr	R
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF (двоичные)	#0001...#FFFF (двоичные)	–
Регистры данных	DR0...DR15	–	
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047,IR0...-2048...+2047, IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(++) ,(-)IR0...,-(-)IR15		

**Описание**

По команде /U(432) производится деление двоичного значения числа без знака, находящегося в Dd, на двоичное значение числа без знака, находящегося в Dr. Частное от деления заносится в R, а остаток - в R+1.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения деления содержание R равно нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующего слова R равен 1. OFF в других случаях.

**Меры предосторожности**

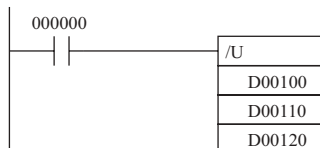
В случае, когда в результате выполнения деления содержание R+1 равно нулю, определяется ошибка, и флаг ошибки переводится в состояние ON.

Когда в результате выполнения деления содержание результирующего слова R равно 0000 (шестн.), флаг равенства переводится в состояние ON.

Если в результате выполнения команды старший бит результирующего слова и R равен 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится деление содержания слова D00100 на содержание слова D00110, в виде двоичных значений четырехзначных чисел без знака. Частное от деления записывается в D00120, а остаток – в D00121.



## 3-11-26 Команда деления двойных двоичных данных без знака DOUBLE UNSIGNED BINARY DIVIDE: /U(433)

**Назначение**

По команде /U(433) осуществляется деление восьмизначных шестнадцатеричных данных без знака и/или констант.

**Символ релейно-контактной схемы**

/UL(433)	
Dd	Dd: Первое слово делимого
Dr	Dr: Первое слово делителя
R	R: Первое слово результата (частного)

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	/U(433)
	Выполняется один раз для дифференцирования вверх	@ /U(433)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

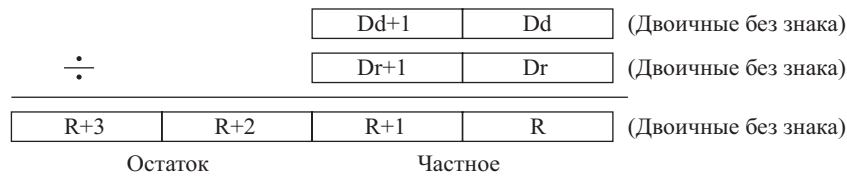
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Dd	Dr	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	CIO 0000...CIO 6140	
Рабочая область	W000...W510	W000...W508	
Область удержания бита (Holding bit area)	H000...H510	H000...H508	
Область вспомогательных битов	A000...A958	A448...A956	
Область таймера	T0000...T4094	T0000...T4092	
Область счетчика	C0000...C4094	C0000...C4092	
Область DM	D00000...D32766	D00000...D32764	
Область EM, не содержащая банков	E00000...E32766	E00000...E32764	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	En_00000...En_32764 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#00000000...#FFFF FFFF (двоичные)	#00000001...#FFFF FFFF (двоичные)	–
Регистры данных	–		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15		

**Описание**

По команде /UL(433) осуществляется деление двоичных значений чисел без знака, находящихся в Dd, Dd+1 на двоичные значения чисел, находящихся в Dr, Dr+1. Частное от деления выводится в слова R, R+1, а остаток – в слова R+2 и R+3.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержание R+1, R равно нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующих слов R+1, R равен 1. OFF в других случаях.

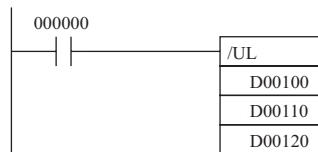
**Меры предосторожности**

В случае, когда содержание остатка Dr, Dr+1 равно нулю, флаг ошибки переводится в состояние ON. Когда в результате выполнения деления содержание результирующих слов R, R+1, равно 00000000 (шести.), флаг равенства переводится в состояние ON.

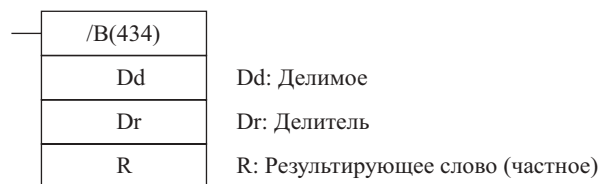
Если в результате выполнения команды старший бит результирующих слов R+1, R равен 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, содержание слов D00100, D00110, делится на содержание слов D00111 и D00112, в виде двоичных значений восьмизначных чисел без знака. Частное от деления записывается в D00121 и D00120, а остаток – в D00123 и D 00122..

**3-11-27 Команда деления двоично-десятичных BCD DIVIDE: /B(434)****Назначение**

По команде /B(434) осуществляется деление двоично-десятичных данных четырехзначных (простого слова) и/или констант.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	/B(434)
	Выполняется один раз для дифференцирования вверх	@ /B(434)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

### Спецификации операндов

Область	Dd	Dr	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6142	
Рабочая область	W000...W511	W000...W510	
Область удержания бита (Holding bit area)	H000...H511	H000...H510	
Область вспомогательных битов	A000...A959	A448...A958	
Область таймера	T0000...T4095	T0000...T4094	
Область счетчика	C0000...C4095	C0000...C4094	
Область DM	D00000...D32767	D00000...D32766	
Область EM, не содержащая банков	E00000...E32767	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#9999 (двоично-десятичные)	#0001...#9999 (двоично-десятичные)	–
Регистры данных	DR0...DR15	–	
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...IR15(++) ,(- -)IR0...,-(- -)IR15		

### Описание

По команде /B(434) осуществляется деление двоично-десятичного числа, находящегося в Dd на двоично-десятичное число, находящееся в Dr. Частное от деления выводится в слово R, а остаток – в слово R+1.

$$\frac{\boxed{\text{Dd}} \text{ (Двоично-десятичное)}}{\boxed{\text{Dr}} \text{ (Двоично-десятичное)}} = \frac{\boxed{\text{R+1}} \text{ (Двоично-десятичные)}}{\boxed{\text{R}} \text{ (Двоично-десятичные)}}$$

Остаток                      Частное

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание Dd выражено не в двоично-десятичном коде. Переводится в состояние ON, когда содержание Dr выражено не в двоично-десятичном коде. Переводится в состояние ON, когда остаток равен нулю. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержание R равно нулю. OFF в других случаях.

### Меры предосторожности

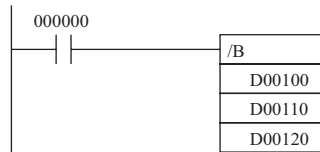
В случае, когда Dd и/или Dr выражены не в двоично-десятичном коде или если остаток R+1 равен нулю, определяется ошибка и флаг ошибки переводится в состояние ON.

Когда в результате выполнения команды содержание результирующего слова R равно 0000 (шестн.), флаг равенства переводится в состояние ON.

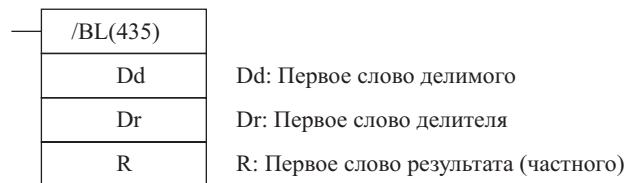
Если в результате выполнения деления старший бит слова R равен 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

В следующем ниже примере, когда СЮ 000000 находится в состоянии ON, содержание слова D00100 делится на содержание слова D00110, в виде четырехзначных двоично-десятичных чисел. Частное от деления записывается в слово D00121, а остаток – в слово D00120.

**3-11-28 Команда деления двойных двоично-десятичных данных DOUBLE BCD DIVIDE: /BL(435)****Назначение**

По команде /BL(435) осуществляется деление восьмизначных двоично-десятичных данных (двойное слово) и/или констант.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	/BL(435)
	Выполняется один раз для дифференцирования вверх	@ /BL(435)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

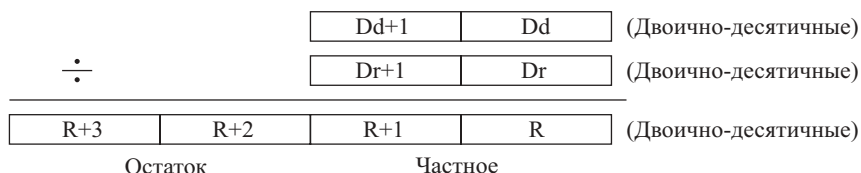
Область	Dd	Dr	R
Область ввода/вывода (область СЮ)	СЮ 0000...СЮ 6142	СЮ 0000...СЮ 6140	
Рабочая область	W000...W510	W000...W508	
Область удержания бита (Holding bit area)	H000...H510	H000...H508	
Область вспомогательных битов	A000...A958	A448...A956	
Область таймера	T0000...T4094	T0000...T4092	
Область счетчика	C0000...C4094	C0000...C4092	
Область DM	D00000...D32766	D00000...D32764	
Область EM, не содержащая банков	E00000...E32766	E00000...E32764	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	En_00000...En_32764 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	00000000...9999999 9 (двоично-десятичные)	00000001...9999999 9 (двоично-десятичные)	–
Регистры данных	–		
Индексные регистры	–		



Область	Dd	Dr	R
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...IR15(++) ,-(-)IR0...,-(-)IR15		

**Описание**

По команде /BL(435) осуществляется деление двоично-десятичных чисел, находящихся в Dd, Dd+1 на числа, находящиеся в Dr, Dr+1. Частное от деления выводится в слова R, R+1, а остаток – в слова R+2 и R+3.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание Dd и/или Dd+1 выражено не в двоично-десятичном коде. Переводится в состояние ON, когда содержание Dr и/или Dr+1 выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.

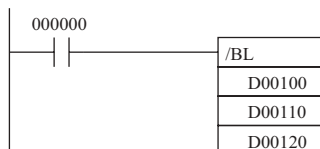
**Меры предосторожности**

В случае, когда Dd, Dd+1 и/или Dr, Dr+1 выражены не в двоично-десятичном коде, или содержание Dr, Dr+1 равно нулю, определяется ошибка и флаг ошибки переводится в состояние ON.

Когда в результате выполнения команды содержание результирующих слов R, R+1 равно 00000000 (шестн.), флаг равенства переводится в состояние ON.

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, содержание слов D00101, D00100 делится на содержание слов D00111 и D00110, в виде восьмизначных двоично-десятичных чисел. Частное от деления записывается в слова D00123 D00122, а остаток – в слова D00121 и D00120.



## 3-12 Команды преобразования

В настоящем разделе приводится описание команд, используемых для преобразования данных.

Команда	Мнемоническое изображение	Код функции	Страница
BCD-TO BINARY	BIN	023	339
DOUBLE BCD-TO- DOUBLE BINARY	BINL	058	340
BINARY-TO-BCD	BCD	024	342
DOUBLE BINARY-TO- DOUBLE BCD	BCDL	059	343
2'S COMPLEMENT	NEG	160	345
DOUBLE 2'S COMPLEMENT	NEGL	161	347
16 – BIT TO 32 – BIT SIGNED BINARY	SIGN	600	349
DATA RECODER	MLPX	076	350
DATA ENCODER	DMPX	077	355
ASCII CONVERT	ASC	086	359
ASCII TO HEX	HEX	162	363
COLUMN TO LINE	LINE	063	367
LINE TO COLUMN	COLM	064	369
SIGNED BCD – TO - BINARY	BINS	470	371
DOUBLE SIGNED BCD – TO - BINARY	BISL	472	374
SIGNED BINARY – TO - BCD	BCDS	471	377
DOUBLE SIGNED BINARY – TO - BCD	BDSL	473	379

## 3-12-1 Команда преобразования двоично-десятичных данных в двоичные данные BCD-TO-BINARY: BIN(023)

**Назначение**

По команде BIN(023) осуществляется преобразование двоично-десятичных данных в двоичные данные.

**Символ релейно-контактной схемы**

—	BIN(023)	
	S	S: Исходное слово
	R	R: Результирующее слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	BIN(023)
	Выполняется один раз для дифференцирования вверх	@ BIN(023)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

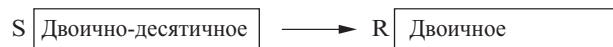
**Спецификации операндов**

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A000...A959	A448...A959
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	

Область	S	R
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	—	
Регистры данных	DR0...DR15	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0... IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-)IR15	

**Описание**

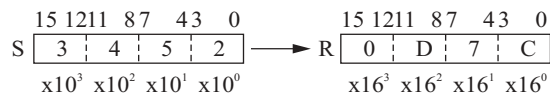
По команде BIN(023) осуществляется преобразование двоично-десятичных данных, находящихся в слове S, в двоичные данные. Результат записывается в слово R.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание S выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержание R равно нулю. OFF в других случаях.
Флаг отрицательного значения		OFF

**Пример**

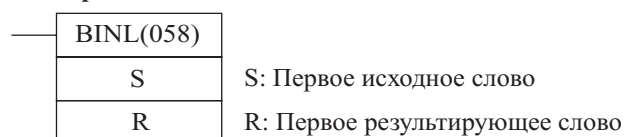
На следующем ниже рисунке приводится пример преобразования двоично-десятичных данных в двоичные данные



### 3-12-2 Команда преобразования двойных двоично-десятичных данных в двойные двоичные данные DOUBLE BCD-TO- DOUBLE BINARY: BINL(058)

**Назначение**

По команде BINL(058) осуществляется преобразование восьмизначных двоично-десятичных данных в восьмизначные шестнадцатеричные данные (32-х битовые двоичные данные).

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	BINL(058)
	Выполняется один раз для дифференцирования вверх	@ BINL(058)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	-	
Регистры данных	-	
Индексные регистры	-	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0... IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15	

**Описание**

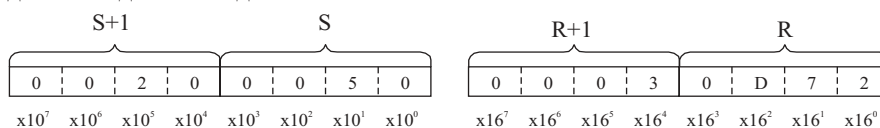
По команде BINL(058) осуществляется преобразование восьмизначных двоично-десятичных данных, находящихся в словах S, S+1, в восьмизначные шестнадцатеричные (32-х битовые двоичные) данные. Результат записывается в слова R, R+1.

**Флаги**

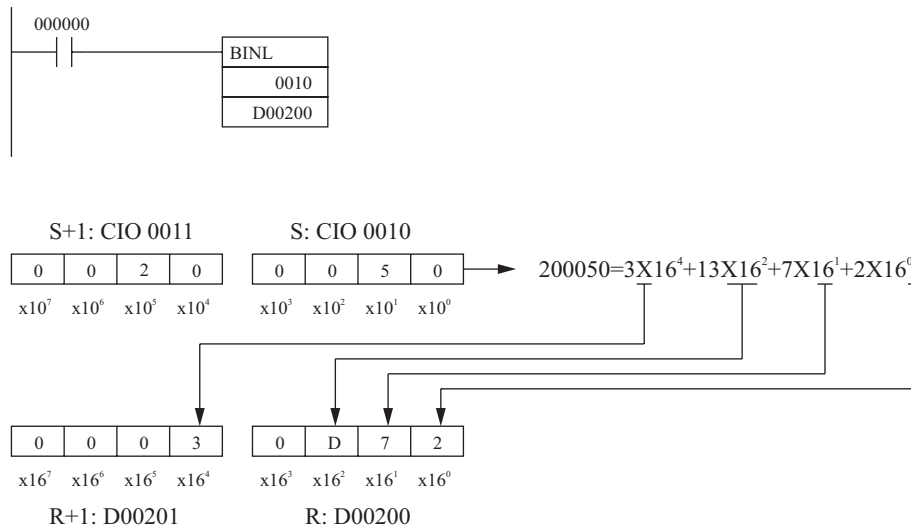
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание S и S+1 выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг отрицательного значения	N	OFF

**Примеры**

На следующем ниже рисунке приводится пример преобразования восьмизначных двоично-десятичных данных в двоичные данные



В следующем ниже примере, когда CIO 000000 находится в состоянии ON, восьмизначное двоично-десятичное число, находящееся в CIO 0010 и CIO 0011, преобразуется в шестнадцатеричное число и затем записывается в слова D00200 и D00201.

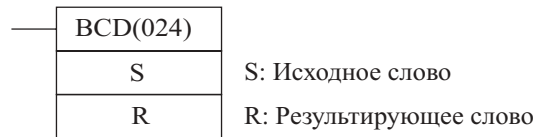


### 3-12-3 Команда преобразования двоичных данных в двоично-десятичные данные BINARY-TO-BCD: BCD(024)

#### Назначение

По команде BCD(024) осуществляется преобразование одного слова двоичных данных в одно слово двоично-десятичных данных.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	BCD(024)
	Выполняется один раз для дифференцирования вверх	@BCD(024)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

##### S: исходное слово

Значение исходного слова S должно находиться в пределах от 0000 до 270F в шестнадцатеричном коде (от 0000 до 9999 в десятичном коде).

#### Спецификации операндов

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A000...A959	A448...A959
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	

Область	S	R
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	-	
Регистры данных	DR0...DR15	
Индексные регистры	-	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0... IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,(-)IR15	

**Описание**

По команде BCD(024) осуществляется преобразование двоичных данных, находящихся в S, в двоично-десятичные данные. Результат записывается в слово R.

**Флаги**

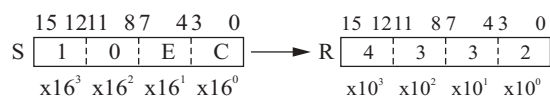
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание S превышает значение 270F (9999 в десятичном коде). OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.

**Меры предосторожности**

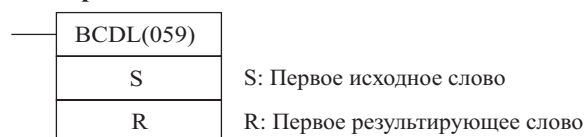
Значение исходного слова S не должно превышать 270F (9999 в десятичном коде).

**Пример**

На следующем ниже рисунке приводится пример преобразования двоичных данных в двоично-десятичные.

**3-12-4 Команда преобразования двойных двоичных данных в двойные двоично-десятичные данные DOUBLE BINARY -TO- DOUBLE BCD: BCDL(059)****Назначение**

По команде BCDL(059) осуществляется преобразование восьмизначных шестнадцатеричных данных (32-х битовые двоичные данные) в восьмизначные двоично-десятичные данные.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	BCDL(059)
	Выполняется один раз для дифференцирования вверх	@ BCDL(059)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****S: Первое исходное слово**

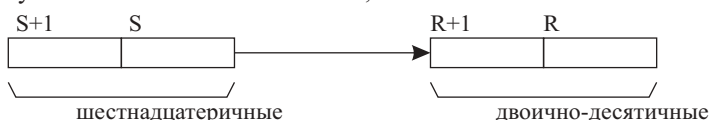
Содержание слов S+1 и S должно находиться в пределах от 00000000 до 05F5 E0FF в шестнадцатеричном коде (от 00000000 до 99999999 в десятичном коде.)

**Спецификации операндов**

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	—	
Регистры данных	—	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0... IR15 ,IR0+(++)... ,IR15+(++) ,(- -)IR0...,(- -)IR15	

**Описание**

По команде BCDL(059) осуществляется преобразование восьмизначных шестнадцатеричных (32-х битовых двоичных) данных, находящихся в словах S, S+1, в восьмизначные двоично-десятичные данные. Результат записывается в слова R, R+1.

**Флаги**

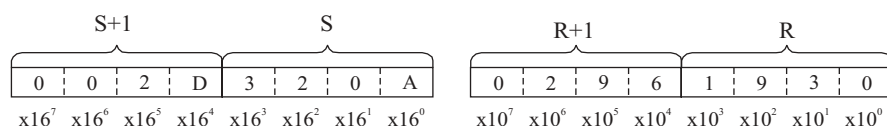
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание S и S+1 превышает значение 05F5 E0FF (99999999 в десятичном коде). OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.

**Меры предосторожности**

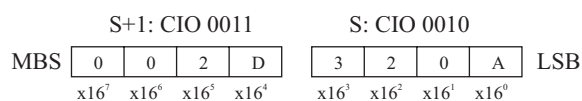
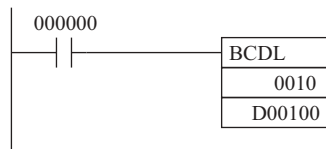
Значение исходных слов S+1 и S не должно превышать 05F5E0FF (99999999 в десятичном коде).

**Примеры**

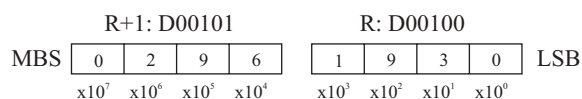
На следующем ниже рисунке приводится пример преобразования двоично-десятичных данных в восьмизначные шестнадцатеричные данные.



В следующем ниже примере, когда СЮ 000000 находится в состоянии ON, шестнадцатеричное число, находящееся в СЮ 0010 и СЮ 0011, преобразуется в двоично-десятичное число и затем записывается в слова D00200 и D00201.



$$2 \times 16^5 + 13 \times 16^4 + 3 \times 16^3 + 2 \times 16^2 + 10 = 2961930$$

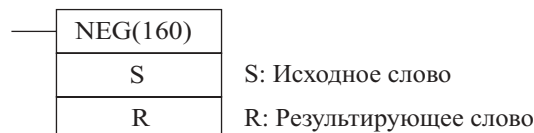


### 3-12-5 Команда вычисления дополнения по модулю 2: 2'S COMPLEMENT: NEG(160)

#### Назначение

По команде NEG(160) осуществляется вычисление дополнения по модулю 2 к одному слову данных, выраженному в шестнадцатеричном коде.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	NEG(160)
	Выполняется один раз для дифференцирования вверх	@ NEG(160)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Спецификации операндов

Область	S	R
Область ввода/вывода (область СЮ)	СЮ 0000...СЮ 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A000...A959	A448...A959
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	



Область	S	R
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#0000...#FFFF (двоичные)	
Регистры данных	DR0...DR15	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0... IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15	

**Описание**

По команде NEG(160) производится вычисление дополнения по модулю 2 к данным в слове S и запись результата в слово R. Суть вычисления дополнения заключается в инвертировании состояния битов слова S и добавлении единицы к результату.

Дополнение по модулю 2  
(дополнение +1)

$\overline{(S)} \longrightarrow (R)$

**Примечание:** Данная операция (инвертирование состояния битов и прибавление единицы) эквивалентна вычитанию содержания S из числа 0000.

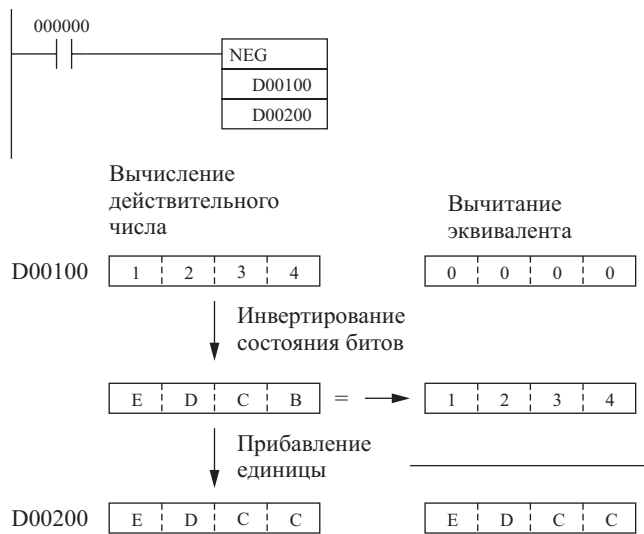
**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда бит 15 результирующего слова равен 1. OFF в других случаях.

**Примечание:** Результат при вычислении дополнения к числу 8000 (шестн.) равен 8000 (шестн.).

**Пример**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, по команде NEG(160) осуществляется вычисление дополнения по модулю 2 содержания слова D00100, и запись результата в слово D00200.

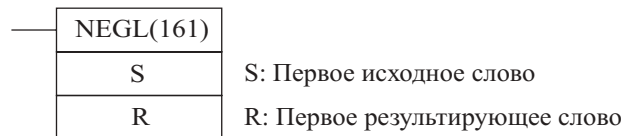


### 3-12-6 Команда вычисления дополнения к двойному слову данных по модулю 2: DOUBLE 2'S COMPLEMENT: NEGL(161)

#### Назначение

По команде NEGL(161) осуществляется вычисление дополнения по модулю 2 к двум словам данных, выраженным в шестнадцатеричном коде.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	NEGL(161)
	Выполняется один раз для дифференцирования вверх	@ NEGL(161)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Спецификации операндов

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	

Область	S	R
Константы	#000000000...#FFFFFFF (двоичные)	
Регистры данных	–	
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0... IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15	

**Примечание:** Слова R и R+1 должны находиться в одной области данных.

**Описание**

По команде NEGL(161) осуществляется вычисление дополнения по модулю 2 к данным в словах S+1 и S, а также запись результата в слова R+1 и R. Суть вычисления дополнения заключается в инвертировании состояния битов слов S+1 и S и добавлении единицы к результату.

Дополнение по модулю 2 (дополнение +1)

$$(\overline{S+1}, \overline{S}) \longrightarrow (R+1, R)$$

**Примечание:** Данная операция (инвертирование состояния битов и прибавление единицы) эквивалентна вычитанию содержания S+1 и S из числа 00000000.

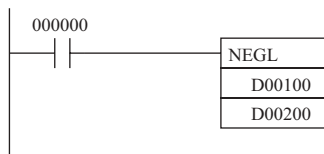
**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен 00000000. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда бит 15 результирующего слова R+1 равен 1. OFF в других случаях.

**Примечание:** Результат при вычислении дополнения к числу 8000 (шестн.) равен 8000 (шестн.).

**Пример**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, по команде NEGL(161) осуществляется вычисление дополнения по модулю 2 содержания слов D00101 и D00100, и запись результата в слова D00201 и D00200.

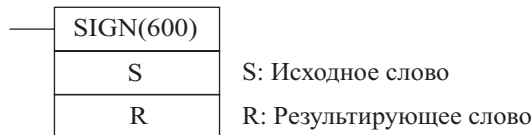


### 3-12-7 Команда преобразования 16-битового числа в его 32-х битовый эквивалент 16- BIT TO 32 - BIT SIGNED BINARY: SIGN(600)

#### Назначение

По команде SIGN(600) осуществляется расширение двоичного числа со знаком в его 32-х битовый эквивалент.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	SIGN(600)
	Выполняется один раз для дифференцирования вверх	@ SIGN(600)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Спецификации операндов

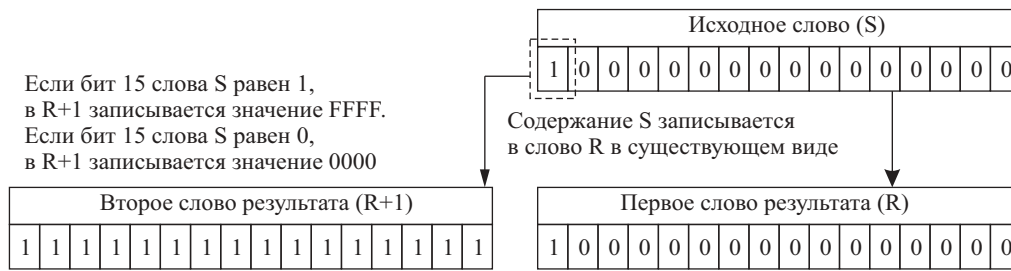
Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6142
Рабочая область	W000...W511	W000...W510
Область удержания бита (Holding bit area)	H000...H511	H000...H510
Область вспомогательных битов	A000...A959	A448...A958
Область таймера	T0000...T4095	T0000...T4094
Область счетчика	C0000...C4095	C0000...C4094
Область DM	D00000...D32767	D00000...D32766
Область EM, не содержащая банков	E00000...E32767	E00000...E32766
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32766 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#0000...#FFFF (двоично-десятичные)	—
Регистры данных	DR0...DR15	—
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0... IR15 ,IR0+(++)...,IR15+(++) ,(-)IR0...,(-)IR15	

**Примечание:** Слова R и R+1 должны находиться в одной области данных.

#### Описание

По команде SIGN(600) осуществляется преобразование 16-ти битового двоичного числа со знаком, находящегося в S, в его 32-х битовый эквивалент. Результат записывается в слова R+1 и R.

Преобразование сопровождается записью содержания S в R и записью FFFF в R+1, если бит 15 слова S равен 1, и записью 0000 в R+1, если бит 15 слова S равен 0.

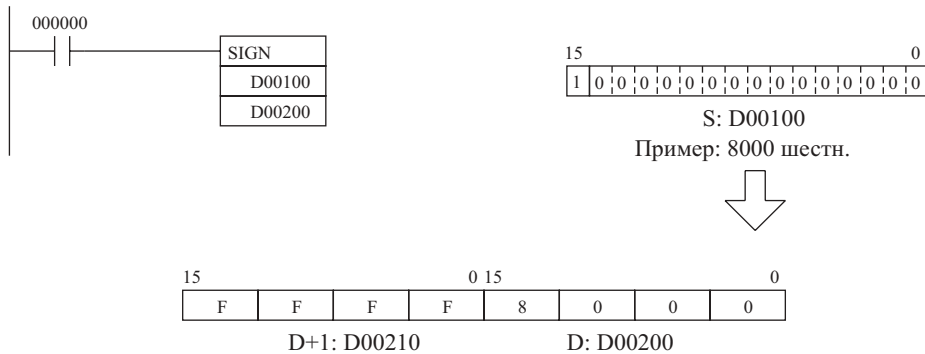


**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен 00000000. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, если бит 15 слова R+1 равен 1. OFF в других случаях.

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, по команде SIGN(600) производится преобразование 16-ти битового двоичного числа со знаком, содержащегося в D00100 (#8000 = -32768 в десятичном коде) в 32-х битовый эквивалент (#FFFF 8000 = - 32768). Результат записывается в слова D00201и D00200.

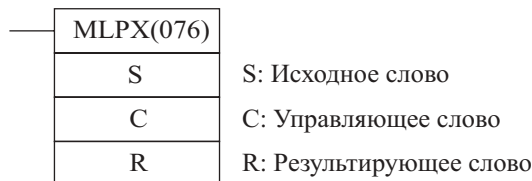


**3-12-8 Команда декодирования данных DATA DECODER: MLPX(076)**

**Назначение**

По команде MLPX(076) осуществляется чтение числового значения указанной цифры (или бита) в исходном слове, перевод в состояние ON соответствующего бита в результирующем слове (или в диапазоне из 16 слов), и перевод в состояние OFF всех прочих битов в результирующем слове (или в диапазоне из 16 слов)

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	MLPX(076)
	Выполняется один раз для дифференцирования вверх	@ MLPX(076)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

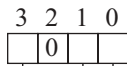
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****S: исходное слово**

Данные в исходном слове указывают расположение бита (битов), который будет переведен в состояние ON.

**C: управляющее слово**

Управляющее слово указывает, будет ли по команде MLPX(076) производиться преобразование 4...16 битов, или преобразование 8...256 битов, количество байтов, подлежащих преобразованию, а также начальную цифру или байт.



Указывает первую цифру/байт.

4...16: 0...3 (цифра от 0 до 3)

8...256: 0 или 1 (байт 0 или 1)

Количество цифр/байтов, подлежащих преобразованию.

4...16: 0 ... 3 (от 1 до 4 цифр)

8...256: 0 или 1 (1 или 2 байта)

Процесс преобразования

0: 4...16 битов (цифра в слове)

1: 8...256 битов (байт в диапазон из 16 слов)

**R: Первое слово результата**

В зависимости от типа процесса преобразования и количества преобразуемых цифр/байтов, в результате может получиться от 1 до 32-х слов. Результирующие слова должны находиться в одной области данных.

**Спецификации операндов**

Область	S	C	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	—	Только указанные значения	—
Регистры данных	DR0...DR15	—	
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0... IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-(-)IR15		

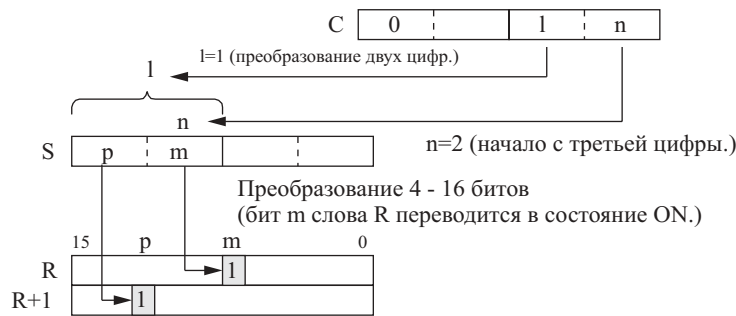
**Описание**

По команде MLPX(076) может производиться преобразование 4→16 битов, или преобразование 8→256 битов. Для выполнения преобразования 4→16 битов устанавливайте цифру старшего разряда C в значение 0, а для выполнения преобразования 8→256 битов – в значение 1.

**Преобразование 4 16 битов**

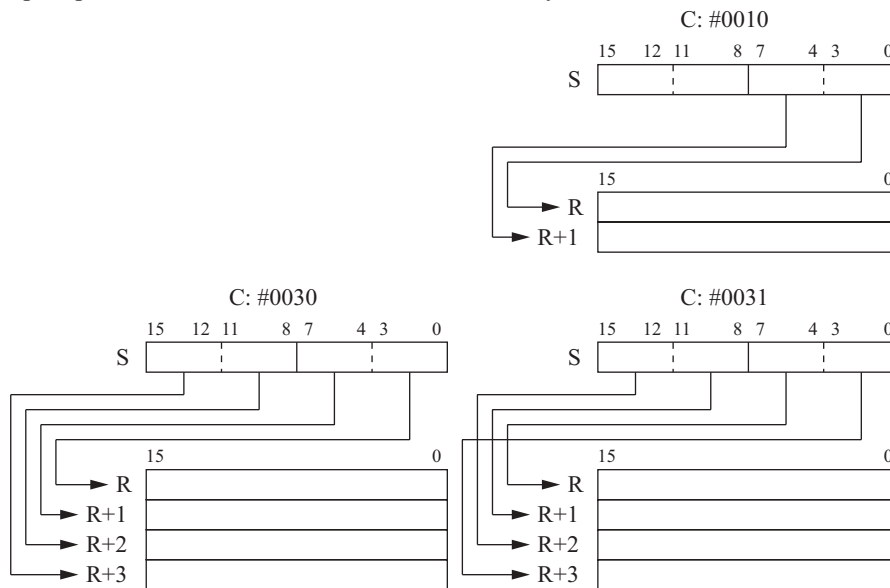
Когда бит старшего разряда слова C равен нулю, по команде MLPX(076) производится чтение значения указанной цифры в слове S (0...F), и перевод в состояние ON соответствующего бита в результирующем слове.

Все остальные биты результирующего слова переводятся в состояние OFF. Таким образом, может производиться преобразование до 4-х цифр.

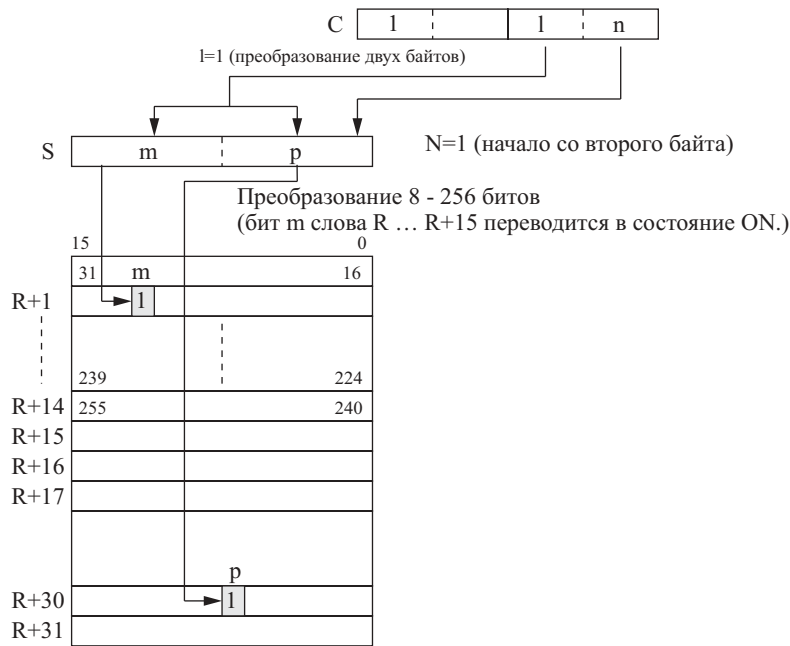


Когда осуществляется преобразование двух или более цифр, по команде MLPX(076) производится чтение цифр в слове S, справа - налево и в обход в обратном направлении к цифре младшего разряда, после чтения цифры старшего разряда.

Следующий ниже рисунок показывает несколько примеров значений слова C и характер производимого преобразования 4...16 битов для каждого из случаев.

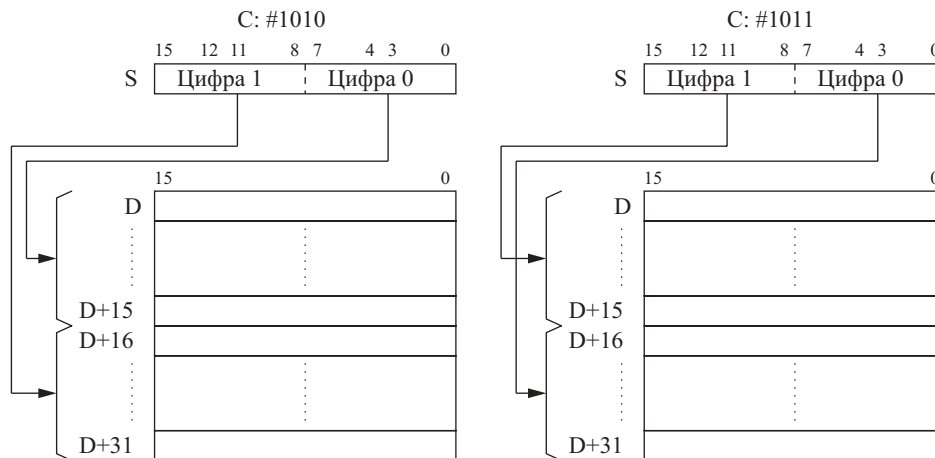
**Преобразование 8 256 битов**

Когда бит старшего разряда слова C равен единице, по команде MLPX(076) производится чтение значения указанного байта в слове S (00...FF), и перевод в состояние ON соответствующего бита в диапазоне из 16-ти результирующих слов. Все остальные биты результирующих слов переводятся в состояние OFF. Таким образом, может производиться преобразование до 2-х байтов.



Когда осуществляется преобразование двух байтов, по команде MLPX(076) производится чтение байтов в слове S, справа - налево и в обход в обратном направлении к байту младшего разряда, если байт старшего разряда (байт 1) был указан в качестве начального байта.

Следующий ниже рисунок показывает несколько примеров значений слова C и характер производимого преобразования 8...256 битов для каждого из случаев.



### Флаги

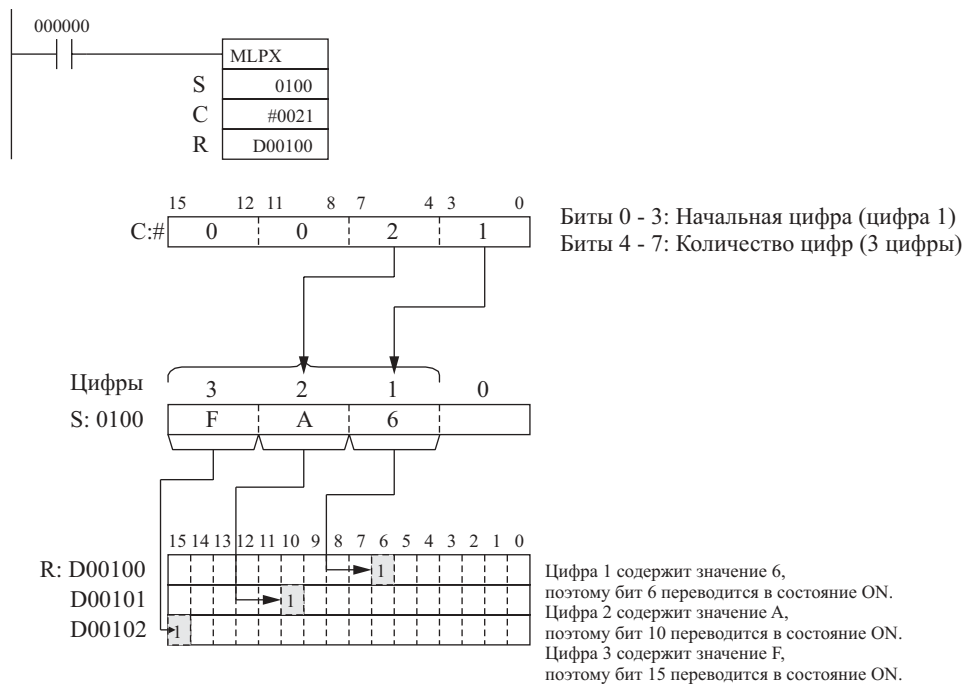
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение C выходит за пределы допустимого диапазона значений. OFF в других случаях.

### Примеры

#### Преобразование 4 16 битов

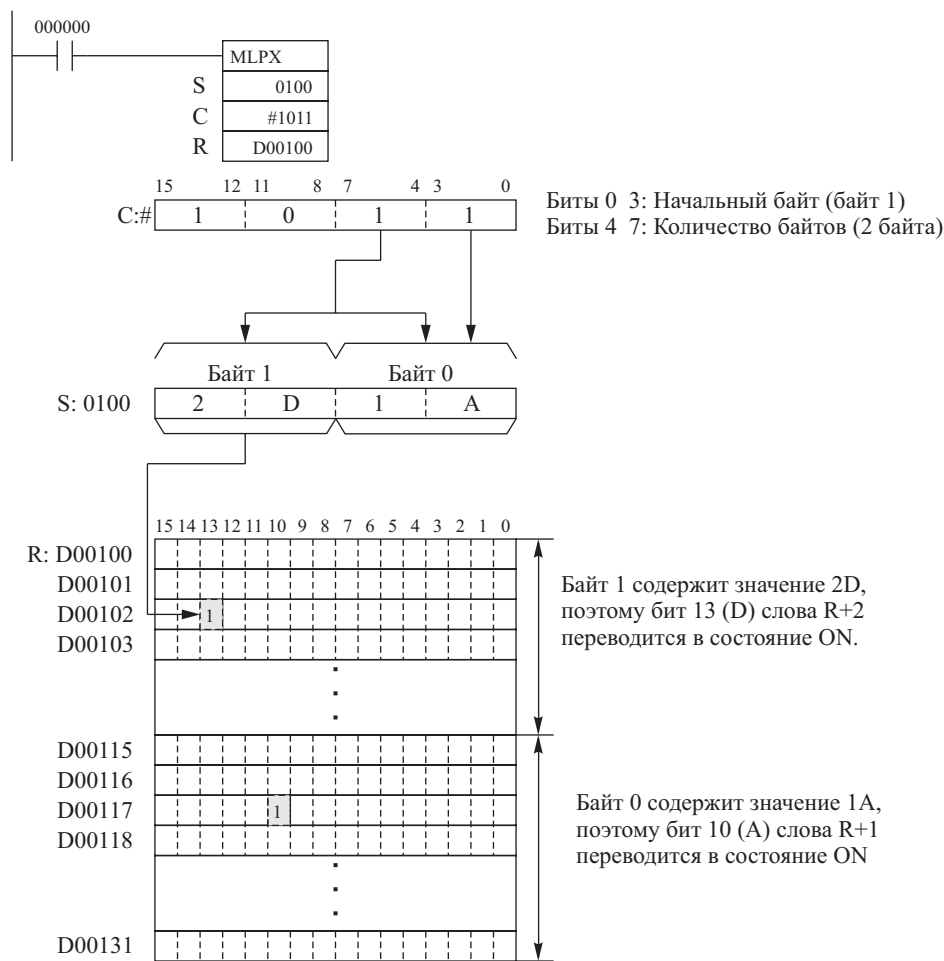
В следующем ниже примере, когда CIO 000000 находится в состоянии ON, по команде MLPX(076) производится преобразование трех цифр слова S, начиная с цифры 1 (второй цифры), как указано в слове C (#0021). Соответствующие биты в D00100, D00101 и D00102 переводятся в состояние ON.



**Преобразование 8...256 битов**

В следующем ниже примере, когда СЮ 000000 находится в состоянии ON, по команде MLPX(076) производится преобразование двух байтов слова S, начиная с байта 1 (байта младшего разряда), как указано в слове С (#1011).

Соответствующие биты в D00100D00115 и D00116...D00131 переводятся в состояние ON.

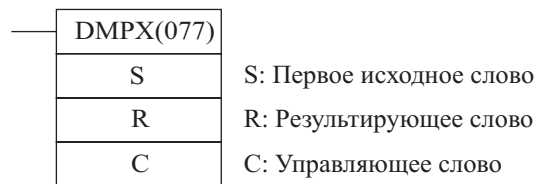


### 3-12-9 Команда кодирования данных DATA ENCODER: DMPX(077)

#### Назначение

По команде DMPX(077) осуществляется нахождение первого или последнего бита, находящегося в состоянии ON, в исходном слове (диапазоне из 16-ти слов), и запись этого значения в указанную цифру (или слово) в результирующем слове.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	DMPX(077)
	Выполняется один раз для дифференцирования вверх	@ DMPX(077)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****S: Первое исходное слово**

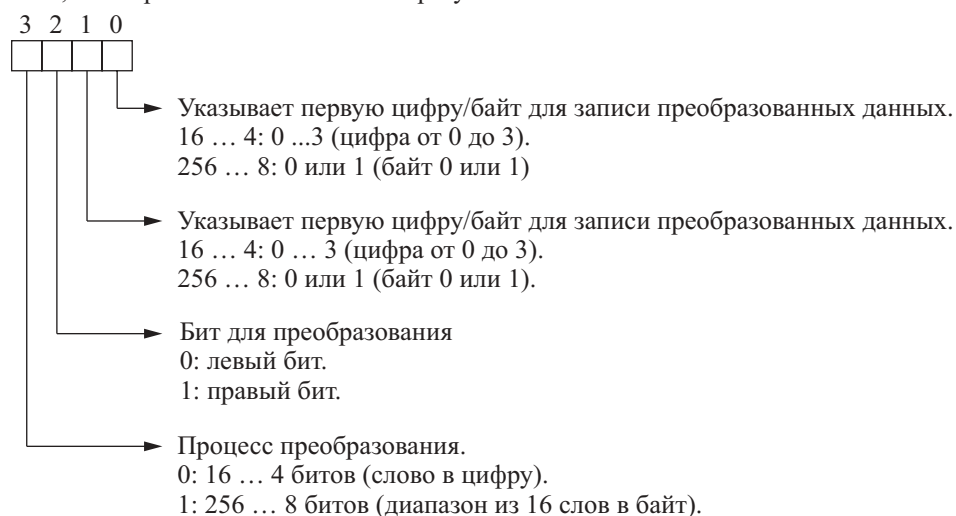
В зависимости от типа процесса преобразования и количества преобразуемых цифр/байтов, может существовать от 1 до 32-х исходных слов. Исходные слова должны находиться в одной области данных.

**R: Результирующее слово**

Расположение битов исходного слова (слов), находящихся в состоянии ON, записывается в цифры/байты слова R, начиная с указанной цифры/байта.

**C: управляющее слово**

Управляющее слово указывает, будет ли по команде DMPX(077) производиться преобразование 16 4 бита, или преобразование 256...8 битов, определяет искомым бит в состоянии ON (старший или младший), количество цифр или байтов, подлежащих преобразованию, а также начальную цифру или байт, с которого начинается запись результата.

**Спецификации операндов**

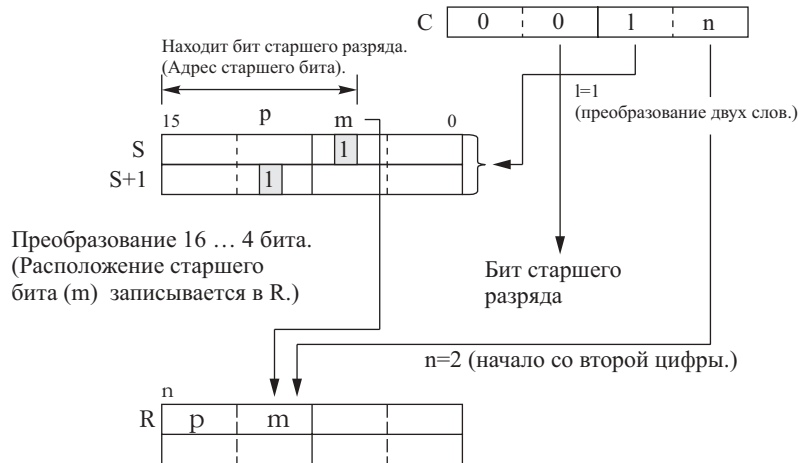
Область	S	R	C
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	A000...A959
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	—	—	Только указанные значения.
Регистры данных	—	DR0...DR15	
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0... IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15		

**Описание**

По команде DMPX(077) может производиться преобразование 16→4 бита, или преобразование 256→8 битов. Для выполнения преобразования 16→4 бита устанавливайте цифру старшего разряда C в значение 0, а для выполнения преобразования 256→8 битов – в значение 1.

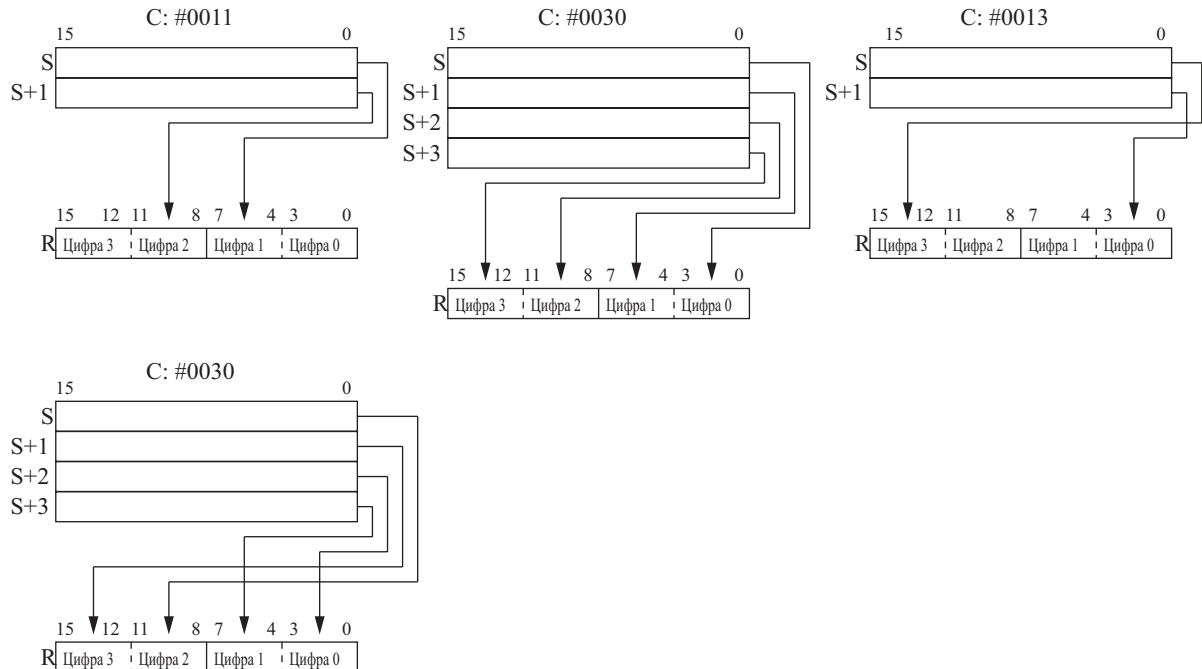
**Преобразование 16 4 бита**

Когда цифра старшего разряда слова C равна нулю, по команде DMPX(077) производится определение расположения старших или младших битов, находящихся в состоянии ON, в исходных словах. Количество исходных слов – до 4. Найденные расположения записываются в слово R, начиная с указанной цифры. (Для нахождения битов старших разрядов устанавливайте третью цифру слова C в значение, равное нулю, а для нахождения битов младших разрядов устанавливайте третью цифру слова C в значение, равное единице).



Когда осуществляется преобразование двух или более цифр, по команде DMPX(077) производится запись значений в цифры слова R справа - налево и, при необходимости, после записи цифры старшего разряда, в обход в обратном направлении к цифре младшего разряда.

Следующий ниже рисунок показывает несколько примеров значений слова C и характер производимого преобразования 16 4 бита для каждого из случаев.

**Преобразование 256 8 битов**

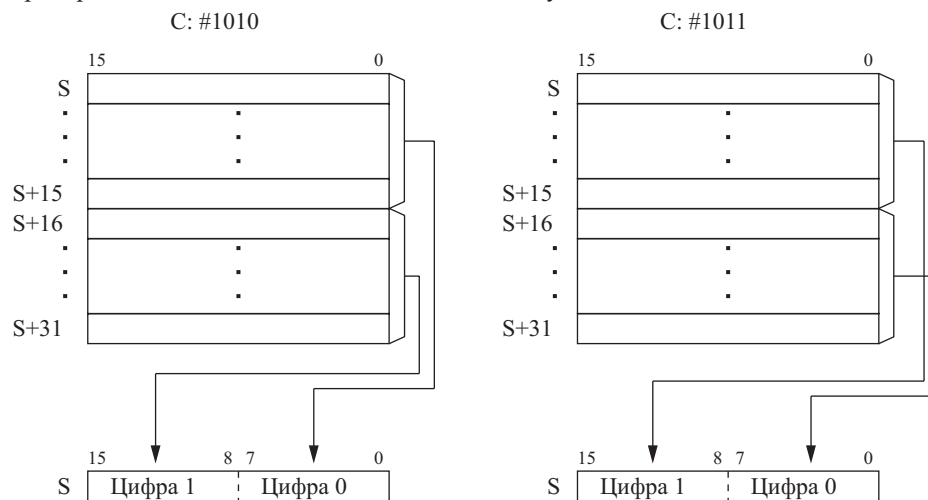
Когда четвертая (старшая) цифра слова C равна единице, по команде DMPX(077) производится определение расположения старших или младших битов, находящихся в состоянии ON, в одном или двух диапазонах исходных слов. Каждый из диапазонов состоит из 16-ти исходных слов. Найденные расположения

записываются в слово R, начиная с указанного байта. (Для нахождения битов старших разрядов устанавливайте третью цифру слова C в значение, равное нулю, а для нахождения битов младших разрядов устанавливайте третью цифру слова C в значение, равное единице).



Когда осуществляется преобразование двух байтов, по команде DMPX(077) производится запись значений в байты слова R, справа - налево и в обход в обратном направлении к байту младшего разряда, если байт старшего разряда (байт 1) был указан в качестве начального байта.

Следующий ниже рисунок показывает несколько примеров значений слова C и характер производимого преобразования 256 8 битов для каждого из случаев.



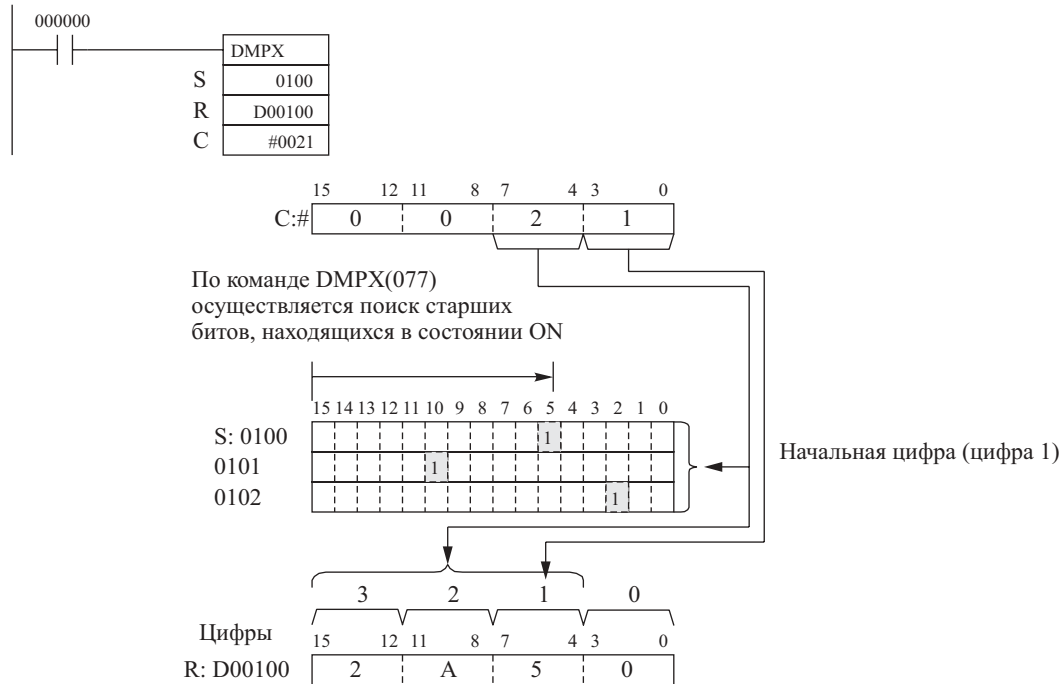
### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда все биты исходного слова (слов) находятся в состоянии OFF. Переводится в состояние ON, когда значение C выходит за пределы допустимого диапазона значений. OFF в других случаях.

### Примеры

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, по команде DMPX(077) производится определение местонахождения старших битов, находящихся в состоянии ON. Поиск осуществля-

ется в словах CIO 0100, CIO 0101 и CIO 0102. Запись найденных адресов производится в три цифры слова R, начиная с цифры 1 (второй цифры), как указано в слове C (#0021).

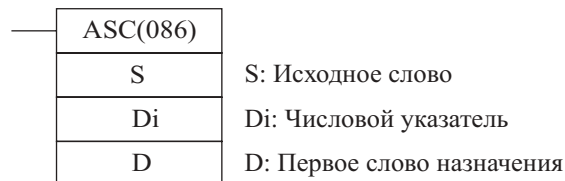


### 3-12-10 Команда преобразования данных в ASCII код ASCII CONVERT: ASC(086)

#### Назначение

По команде ASC(086) осуществляется преобразование 4-х битовых шестнадцатеричных чисел исходного слова в их 8-ми битовый эквивалент в ASCII коде.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	ASC(086)
	Выполняется один раз для дифференцирования вверх	@ ASC(086)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

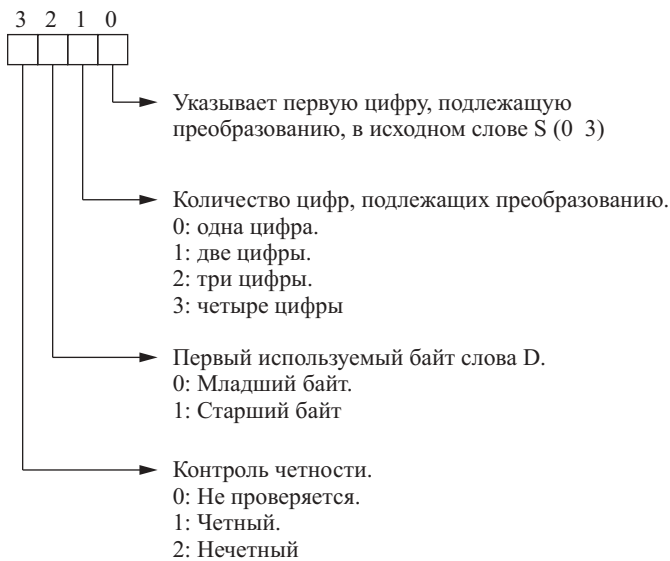
#### Операнды

##### S: Исходное слово

В исходном слове может подвергаться преобразованию до четырех цифр. Эти цифры нумеруются от 0 до 3 справа налево.

##### Di: Числовой указатель

Числовой указатель определяет различные параметры преобразования, как показано на следующем рисунке.

**D: Первое слово назначения**

Данные, преобразованные в ASCII код, записываются в слово (слова) назначения D, начиная с указанного байта. В случае, когда производится преобразование четырех цифр, и в качестве первого байта слова D указан старший байт, требуется четыре слова назначения (D...D+3). Слова назначения должны находиться в одной области данных.

Любой из байтов слова (слов) назначения, который не подвергался перезаписи данными в ASCII коде, остается после выполнения команды неизменным.

**Спецификации операндов**

Область	S	Di	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	—	Только указанные значения.	—
Регистры данных	DR0...DR15	—	
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0... IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15		

**Описание**

Команда ASC(086) обрабатывает содержание слова S в виде четырехзначных шестнадцатеричных чисел. При этом осуществляется преобразование указанной цифры (цифр) в ее (их) восьмибитовый эквивалент в ASCII коде и запись полученного результата в слово (слова) назначения, начиная с заданного байта в слове D.



**Примечание:** Для ознакомления с расширенной таблицей знаков ASCII обратитесь к **Приложению А Руководства по эксплуатации пультов программирования серии CSI (W341)**.

### Контроль четности

Для выполнения контроля ошибок в процессе передачи ASCII данных существует возможность задания проверки четности. Старший бит каждого из ASCII символов будет автоматически корректироваться до четного значения, нечетного значения, либо задается отсутствие проверки на четность. Когда задается отсутствие проверки на четность (0), старший бит всегда будет равен нулю. Когда указывается контроль четности (1), бит старшего разряда корректируется таким образом, что общая сумма битов, равных 1, является четной. Когда задается проверка нечетности (2), старший бит каждого из ASCII символов корректируется таким образом, что общее количество битов, находящихся в состоянии 1, является нечетным. Состояние бита четности не отражается на значении символа ASCII кода.

Пример контроля четности.

При задании контроля четности, выраженное в ASCII коде значение «31» (00110001), будет равно «B1» (10110001: бит контроля четности переводится в состояние ON для получения четного числа битов, находящихся в состоянии ON.) Выраженное в ASCII коде значение «36» (00110110) будет равно «36» (00110110: бит контроля четности остается в состоянии OFF, т.к. количество битов, равных 1, уже является четным.)

Пример контроля нечетности.

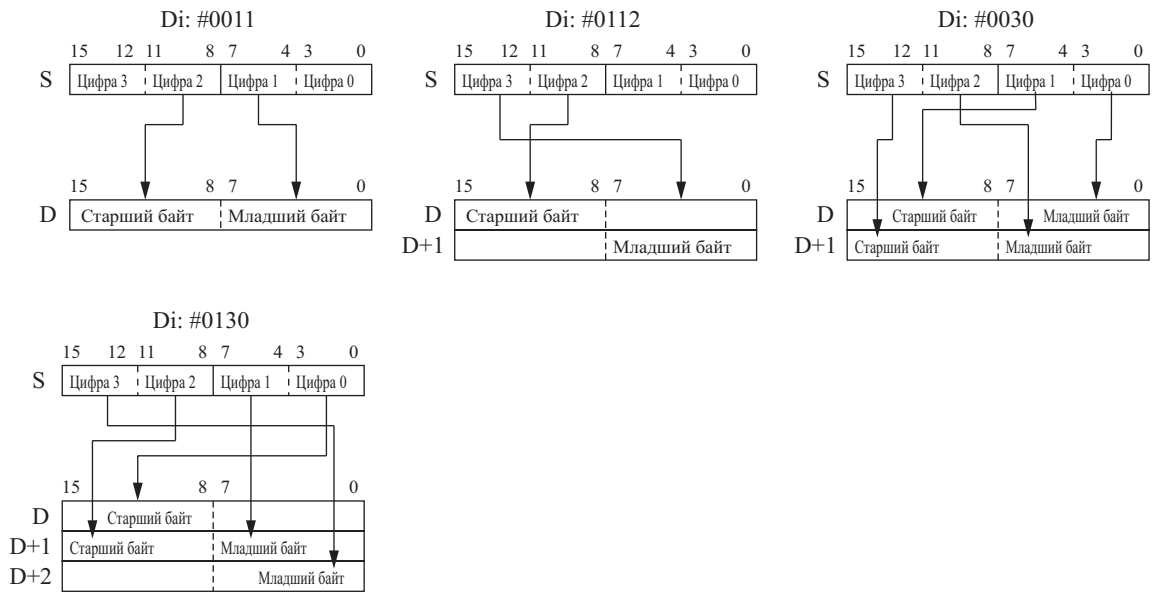
При задании контроля нечетности, выраженное в ASCII коде значение «36» (00110110), будет равно «B6» (10110110: бит контроля четности переводится в состояние ON для получения нечетного числа битов, находящихся в состоянии ON.) Выраженное в ASCII коде значение «46» (01000110) будет равно «46» (01000110: бит контроля четности остается в состоянии OFF, т.к. количество битов, равных 1, уже является нечетным.)

Пример числового указателя  $D_i$

При выполнении преобразования двух или более цифр, по команде ASC(086) производится чтение байтов в слове  $S$  справа налево и затем, если это необходимо в обход в обратном направлении к байту младшего разряда.

На следующем ниже рисунке приводятся несколько примеров значений  $D_i$  и характер выполняемого при этом преобразования.



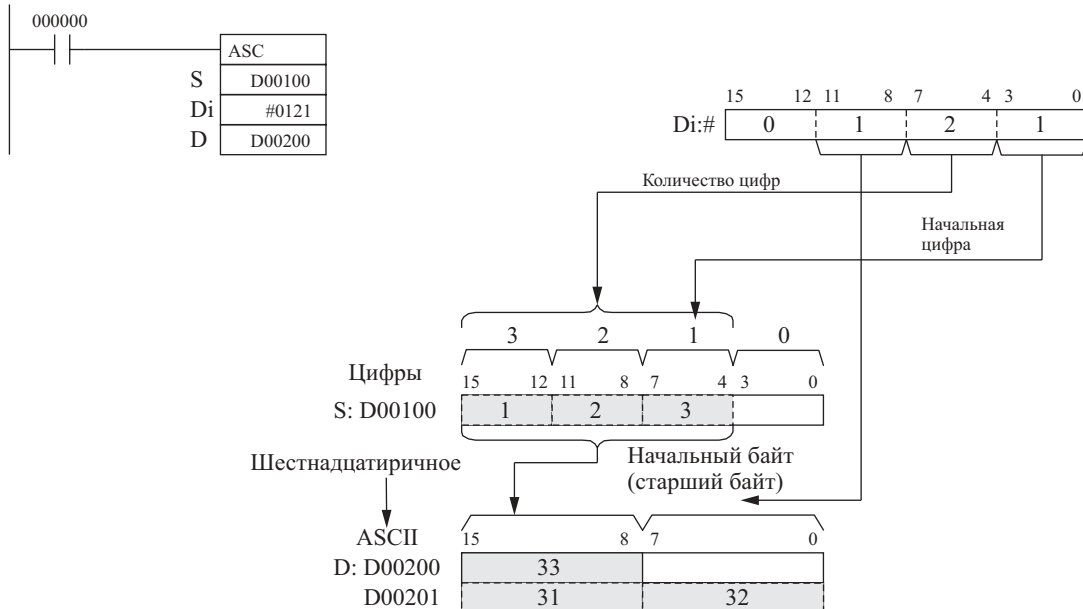


**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение Di выходит за пределы допустимого диапазона значений. OFF в других случаях.

**Пример**

В следующем ниже примере, когда СЮ 000000 находится в состоянии ON, по команде ASC(086) производится преобразование трех шестнадцатеричных цифр, находящихся в D00100 (начиная с цифры 1) в их ASCII эквивалент. Результат записывается в D00200 и D00201, начиная с бита старшего разряда в D00200. В данном случае числовой указатель # 0121 указывает отсутствие проверки на четность, начальным байтом (при записи) указывается старший байт, количество читаемых цифр равно 3, и начальной цифрой при записи является цифра 1.



## 3-12-11 Команда преобразования ASCII данных в шестнадцатеричные данные ASCII TO HEX: HEX(162)

**Назначение**

По команде HEX(162) осуществляется преобразование до 4 байтов ASCII данных в их эквивалент в шестнадцатеричном коде и запись результата в заданное слово назначения.

**Символ релейно-контактной схемы**

—	HEX(162)	
	S	S: Первое исходное слово
	Di	Di: Числовой указатель
	D	D: Слово назначения

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	HEX(162)
	Выполняется один раз для дифференцирования вверх	@ HEX(162)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

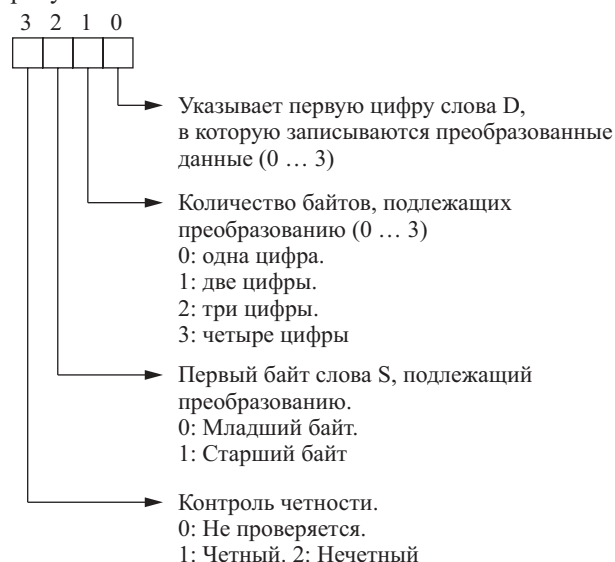
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****S: Первое исходное слово**

Содержание исходных слов обрабатывается в виде ASCII данных. Может использоваться до трех исходных слов. (Три исходных слова потребуются, если осуществляется преобразование четырех байтов и старший байт указан в качестве первого байта в слове S.) Исходные слова должны находиться в одной области данных.

**Di: Числовой указатель**

Числовой указатель определяет различные параметры преобразования, как показано на следующем рисунке.

**D: Первое слово назначения**

Данные, преобразованные в шестнадцатеричный код, записываются в слово назначения D справа налево, начиная с указанной первой цифры. Любая из цифр слова назначения, которая не подвергалась перезаписи преобразованными данными, остается после выполнения команды неизменной.

### Спецификации операндов

Область	S	Di	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	–	Только указанные значения.	–
Регистры данных	–	DR0...DR15	–
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0... IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-(-)IR15		

### Описание

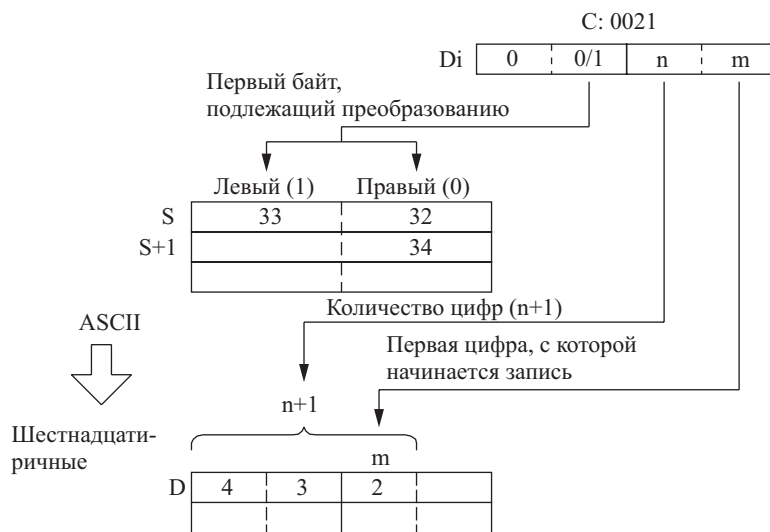
Команда HEX(162) обрабатывает содержание слова (слов) S в виде данных в ASCII коде, выражающих шестнадцатеричные данные (от 0 до 9 и от A до F). При этом осуществляется преобразование указанного количества байтов в шестнадцатеричные данные и запись полученного результата в слово назначения, начиная с заданной цифры.

В случае, когда исходные слова содержат данные в ASCII коде, которые не являются эквивалентом шестнадцатеричных данных, определяется ошибка. В следующей ниже таблице приводятся шестнадцатеричные числа и их эквиваленты в ASCII коде (включая биты контроля четности).

Шестнадцатеричные числа (4 бита)	ASCII эквиваленты (2 шестнадцатеричные цифры)
0...9	30...39
A...F	41...46

**Примечание:** Для ознакомления с расширенной таблицей знаков ASCII обратитесь к *Приложению А Руководства по эксплуатации пультов программирования серии CS1 (W341)*.

На следующем ниже рисунке показано действие команды HEX(162), когда Di = 0021.



**Контроль четности**

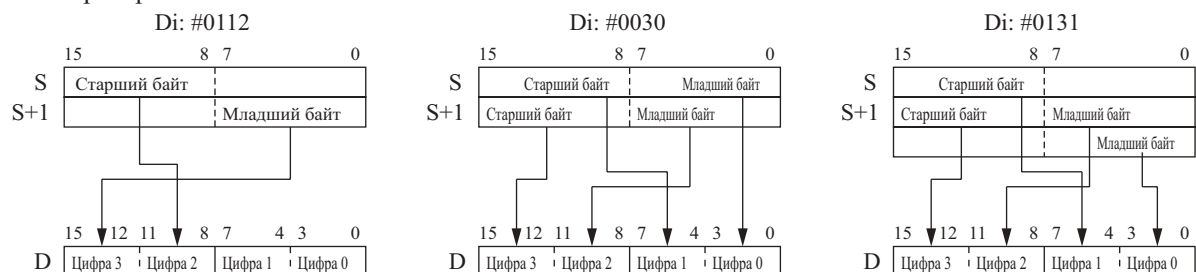
Для контроля ошибок в процессе передачи ASCII данных существует возможность задания проверки четности. Старший бит каждого из байтов является битом контроля четности. Когда задается отсутствие контроля четности, этот бит всегда будет равен нулю. При задании контроля на четность, состояние бита контроля четности дополняет число битов, равных 1, до четного числа. При задании контроля на нечетность, состояние бита контроля четности дополняет число битов, равных 1, до нечетного числа. В следующей таблице приводятся действия команды HEX(162) для каждой из установок для контроля четности.

Установка для контроля четности (старшая цифра $D_i$ )	Функционирование команды HEX(162)
Отсутствие проверки четности (0)	Команда HEX(162) выполняется только тогда, когда бит проверки четности в каждом байте равен нулю. В случае, когда этот бит не равен нулю, определяется ошибка.
Проверка на четность (1)	Команда HEX(162) выполняется только тогда, когда в каждом из байтов количество битов, находящихся в состоянии ON, равно четному числу. В случае, когда байт содержит четное количество битов, равных 1, определяется ошибка.
Проверка на нечетность (2)	Команда HEX(162) выполняется только тогда, когда в каждом из байтов количество битов, находящихся в состоянии ON, равно нечетному числу. В случае, когда байт содержит нечетное количество битов, равных 1, определяется ошибка.

**Пример числового указателя  $D_i$**

При выполнении преобразования двух или более цифр, по команде HEX(162) производится запись преобразованных данных в слово назначения справа налево и затем, если это необходимо, в обход в обратном направлении к байту младшего разряда.

На следующем ниже рисунке приводятся несколько примеров значений  $D_i$  и характер выполняемого при этом преобразования.



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда ASCII данные содержат ошибку четности. Переводится в состояние ON, когда ASCII данные в исходном слове не являются эквивалентом шестнадцатеричного числа. Переводится в состояние ON, когда значение Di выходит за пределы допустимого диапазона значений. OFF в других случаях.

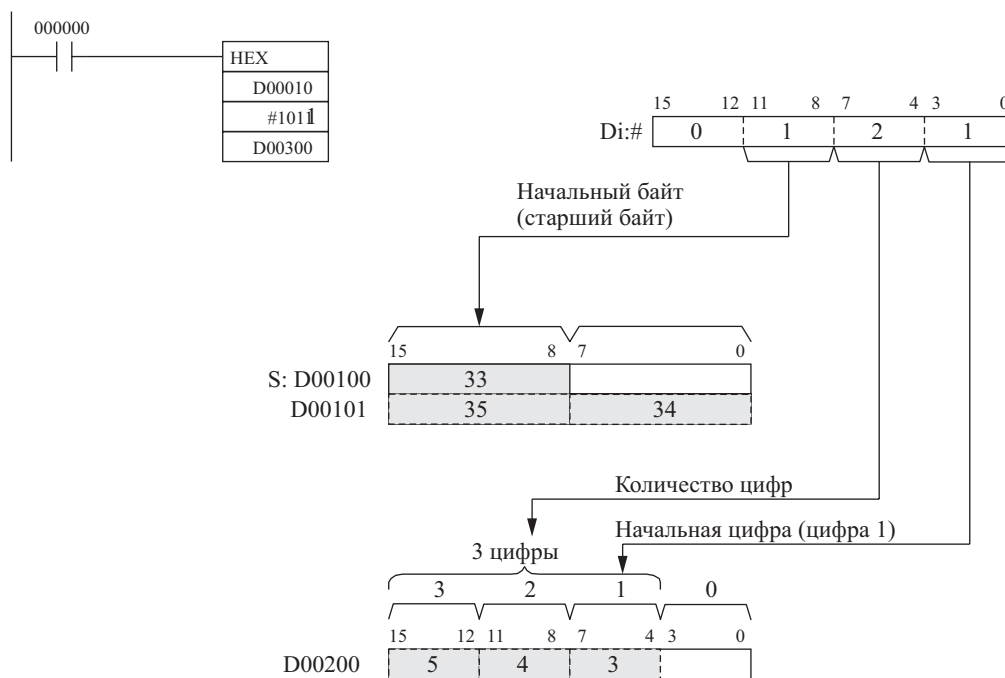
**Меры предосторожности**

В случае, когда ASCII данные содержат ошибку четности, ASCII данные в исходном слове не являются эквивалентом шестнадцатеричного числа, или когда значение Di выходит за пределы допустимого диапазона значений, определяется ошибка и флаг ошибки переводится в состояние ON.

**Примеры**

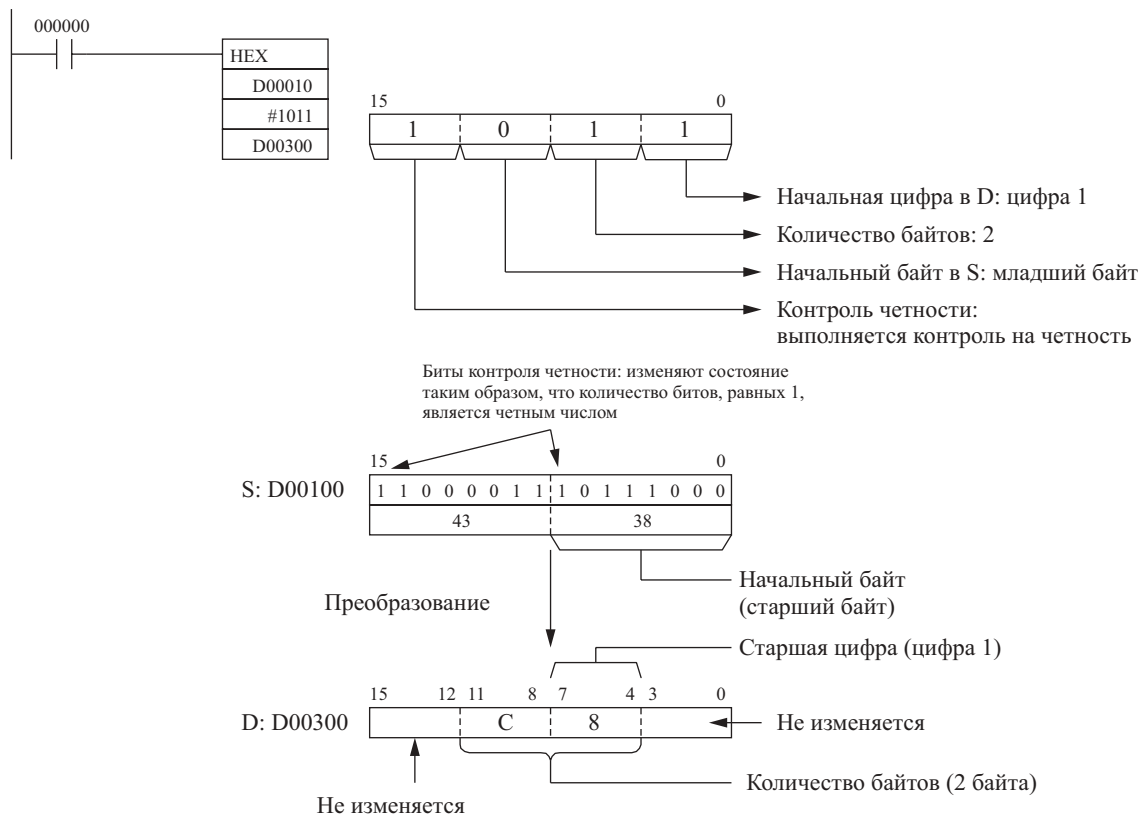
В следующем ниже примере, когда CIO 000000 находится в состоянии ON, по команде HEX(162) производится преобразование ASCII данных, находящихся в D00100 и D00101, согласно установкам в числовом указателе. (Di=#0121 задает отсутствие контроля четности, начальный байт при чтении равен старшему байту, количество читаемых байтов равно 3, начальной цифрой при записи является цифра 1.)

По команде HEX(162) производится преобразование трех байтов ASCII данных (трех цифр), начиная со старшего байта в D00100, в шестнадцатеричный эквивалент, и запись результата в D00200, начиная с цифры 1.



В следующем ниже примере, когда CIO 000000 находится в состоянии ON, по команде HEX(162) производится преобразование ASCII данных, находящихся в D00010 начиная с младшего байта, в шестнадцатеричный эквивалент, и запись результата в D00300, начиная с цифры 1.

Числовой указатель #1011 задает выполнение контроля на четность, начальным байтом при чтении является младший байт, количество читаемых байтов равно 2, начальной цифрой при записи является цифра 1.



### 3-12-12 Команда преобразования данных из колонки в строку COLUMN TO LINE: LINE (063)

#### Назначение

По команде LINE (063) осуществляется преобразование колонки битов из 16 последовательных слов (одинаковый номер бита в 16-ти последовательных словах), в 16 битов слова назначения.

#### Символ релейно-контактной схемы

LINE(063)	
S	S: Первое исходное слово
N	N: Номер бита
D	D: Слово назначения

#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	LINE (063)
	Выполняется один раз для дифференцирования вверх	@ LINE (063)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

##### S: Первое исходное слово

Указывает первое исходное слово. Исходные слова S и S+15 должны находиться в одной области данных.

##### N: Номер бита

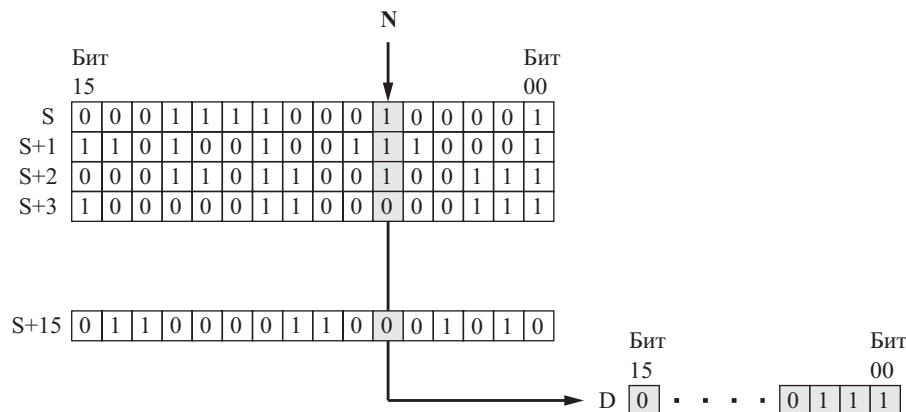
Указывает номер битов (0000...000F или &0...&15), подлежащих чтению в исходных словах.

### Спецификации операндов

Область	S	N	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6128	CIO 0000...CIO 6143	
Рабочая область	W000...W496	W000...W511	
Область удержания бита (Holding bit area)	H000...H496	H000...H511	
Область вспомогательных битов	A000...A944	A000...A959	A448...A959
Область таймера	T0000...T4080	T0000...T4095	
Область счетчика	C0000...C4080	C0000...C4095	
Область DM	D00000...D32752	D00000...D32767	
Область EM, не содержащая банков	E00000...E32752	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32752 (n = 0...C)	En_00000...En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	–	#0000...000F (двоичные) &0...&15 (или)	–
Регистры данных	–	DR0...DR15	
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров		,IR0...IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0... IR15 ,IR0+(++)...IR15(++) ,(- -)IR0...,-(- -)IR15	

### Описание

По команде LINE(063) осуществляется чтение 16-ти битов с номером N из 16 последовательных слов от S до S+15 и запись этих битов в слово назначения D. Бит N слова S+m записывается в бит m слова D, т.е. бит N слова S записывается в бит 00, а бит N S+15 записывается в бит 15 слова D.

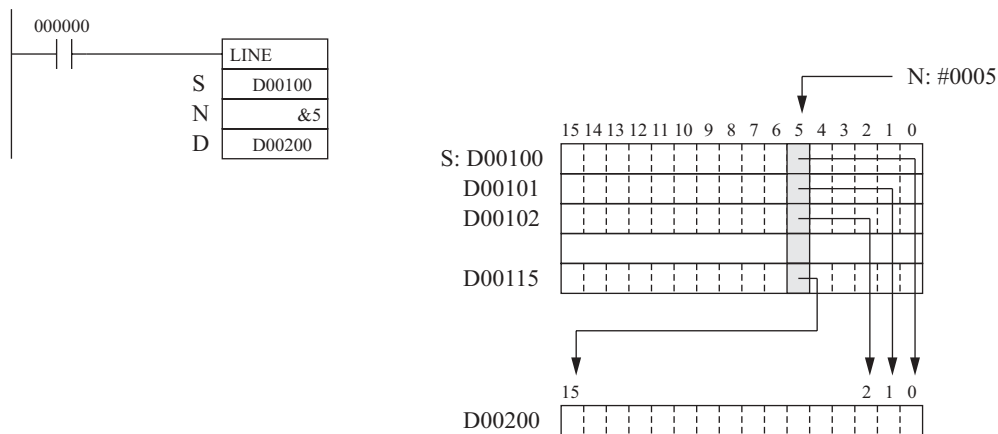


### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение N выходит за пределы допустимого диапазона значений от 0000 до 000F. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды значение D равно 0000. OFF в других случаях.

### Примеры

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, по команде LINE(063) производится чтение бита 5 из слов D00100...D00115 и запись шестнадцати битов в слово D00200.

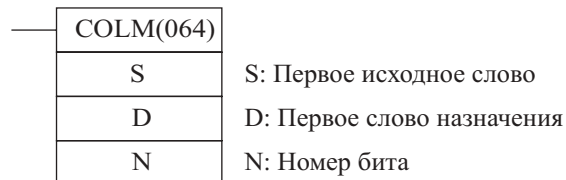


### 3-12-13 Команда преобразования данных из строки в колонку LINE TO COLUMN: COLM (064)

#### Назначение

По команде COLM (064) осуществляется преобразование строки из 16 битов в колонку битов в 16-ти последовательных словах назначения (одинаковый номер бита в 16-ти последовательных словах).

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	COLM (064)
	Выполняется один раз для дифференцирования вверх	@ COLM (064)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

##### D: Первое слово назначения

Указывает первое слово назначения. Слова D и D+15 должны находиться в одной области данных.

##### N: Номер бита

Указывает номер битов (0000...000F или &0...&15), подлежащих перезаписи данными исходного слова.

#### Спецификации операндов

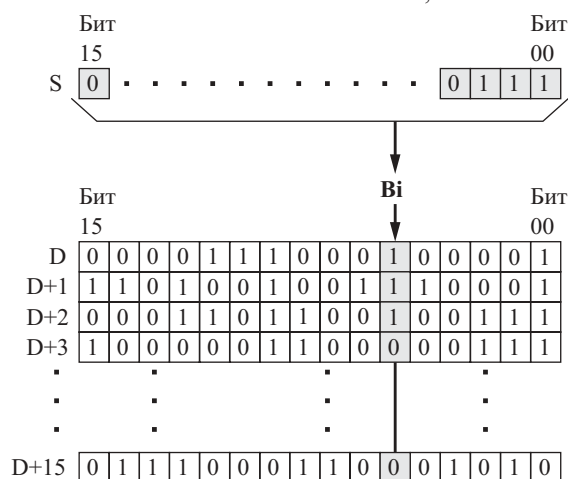
Область	S	N	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6128	CIO 0000...CIO 6143
Рабочая область	W000...W511	W000...W496	W000...W511
Область удержания бита (Holding bit area)	H000...H511	H000...H496	H000...H511
Область вспомогательных битов	A000...A959	A448...A944	A000...A959
Область таймера	T0000...T4095	T0000...T4080	T0000...T4095
Область счетчика	C0000...C4095	C0000...C4080	C0000...C4095
Область DM	D00000...D32767	D00000...D32752	D00000...D32767
Область EM, не содержащая банков	E00000...E32767	E00000...E32752	E00000...E32767



Область	S	N	D
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32752 (n = 0...C)	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...FFFF (двоичные)	–	#0000...000F (двоичные) или &0...&15
Регистры данных	DR0...DR15	–	DR0...DR15
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0... IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15		

**Описание**

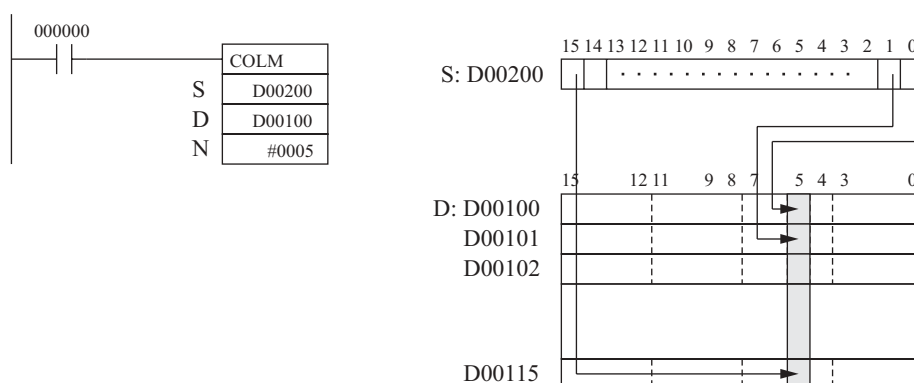
По команде COLM(064) осуществляется чтение 16-ти битов в слове S и запись 16-ти битов с номером N в 16 последовательных слов назначения от D до D+15. Бит m слова S записывается в бит N слова D+m, т.е. бит 00 слова S записывается в бит N, а бит 15 слова S записывается в бит N слова D+15.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение N выходит за пределы допустимого диапазона значений от 0000 до 000F. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды бит N в словах от D до D+15 равен 1. OFF в других случаях.

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, по команде COLN(064) производится чтение шестнадцати битов в слове D00200 (биты 00...15) и запись данных в бит 5 слов D00100...D00115.

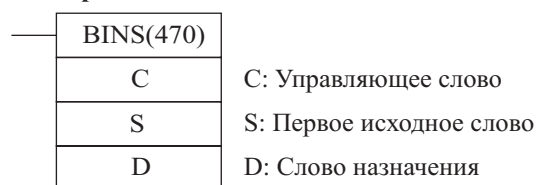


### 3-12-14 Команда преобразования двоично-десятичных данных со знаком в двоичные данные со знаком SIGNED BCD-TO-BINARY: BINS(470)

#### Назначение

По команде BINS(470) осуществляется преобразование одного слова двоично-десятичных данных со знаком в одно слово двоичных данных со знаком.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	BINS(470)
	Выполняется один раз для дифференцирования вверх	@ BINS(470)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

##### C: Управляющее слово

Указывает формат двоично-десятичных данных со знаком. Диапазон значений C- от 0000 до 0003.

#### Спецификации операндов

Область	C	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		

Область	C	S	D
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#0003 (двоичные)	-	
Регистры данных	DR0...DR15		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15-2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0... IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-( -)IR15		

### Описание

По команде BINS(470) осуществляется преобразование двоично-десятичных данных со знаком в двоичные данные со знаком. В начале проверяется формат двоично-десятичных данных со знаком и диапазон значений слова S, на соответствие данным управляющего слова (C).

Если данные в исходном слове верны, двоично-десятичные данные слова S преобразуются в двоичные данные со знаком и записываются в слово D. Если исходные данные некорректны, флаг ошибки переводится в состояние ON, и команда не выполняется.



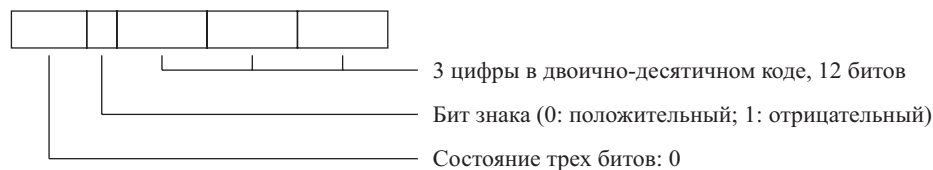
Когда преобразуемые данные являются отрицательным числом, они записываются в виде дополнения по модулю 2 к результирующему числу, и флаг отрицательного значения переводится в состояние ON. Для определения абсолютного значения отрицательного двоичного числа может использоваться команда NEG(160). Для ознакомления с деталями обратитесь к разделу 2-12-5 «Команда вычисления дополнения по модулю 2: NEG(160)».

Значение исходных данных, равное «-0» будет обрабатываться как 0, и не приведет к определению ошибки. Кроме того, когда C=0000, значение битов 13...15 слова S не проверяется.

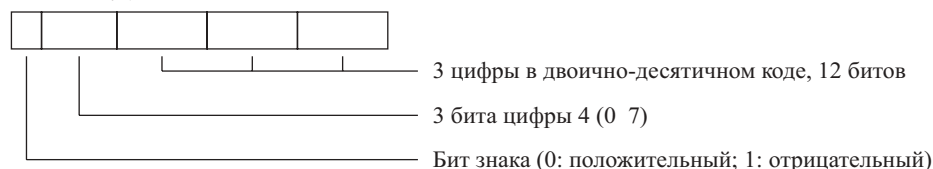
**Примечание:** Некоторые Специальные модули подают на выход двоично-десятичные данные со знаком. Обработка таких данных, связанная с вычислениями, обычно упрощается, если предварительно эти данные преобразовать в двоичные данные со знаком, используя команду BINS(470).

Управляющее слово определяет формат двоично-десятичных данных со знаком, как показано ниже.

**C=0000 (Диапазон изменения входных данных: от -999 до 999 в двоично-десятичном коде)**



**C=0001 (Диапазон изменения входных данных: от -7999 до 7999 в двоично-десятичном коде)**



**C=0002 (Диапазон изменения входных данных: от -999 до 9999 в двоично-десятичном коде)**



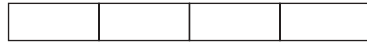
3 цифры в двоично-десятичном коде, 12 битов

0 ... 9: Четвертая цифра в двоично-десятичном коде.

F: Отрицательное число (-)

A ... E: ошибка

**C=0003 (Диапазон изменения входных данных: от -1999 до 9999 в двоично-десятичном коде)**



3 цифры в двоично-десятичном коде, 12 битов

0 ... 9: Четвертая цифра в двоично-десятичном коде.

A: Отрицательное число (-1)

F: Отрицательное число (-)

B ... E: ошибка.

В следующей ниже таблице показаны возможные значения для каждого из двоично-десятичных форматов, и соответствующие двоичные числа со знаком.

Установка	Двоично-десятичные значения со знаком	Двоичные значения со знаком
C=0000	-999...-1 и 0...999	FC19...FFFF и 0000...03E7
C=0001	-7999...-1 и 0...7999	E0C1...FFFF и 0000...1F3F
C=0002	-999...-1 и 0...9999	FC19...FFFF и 0000...270F
C=003	-1999...-1 и 0...9999	F831...FFFF и 0000...270F

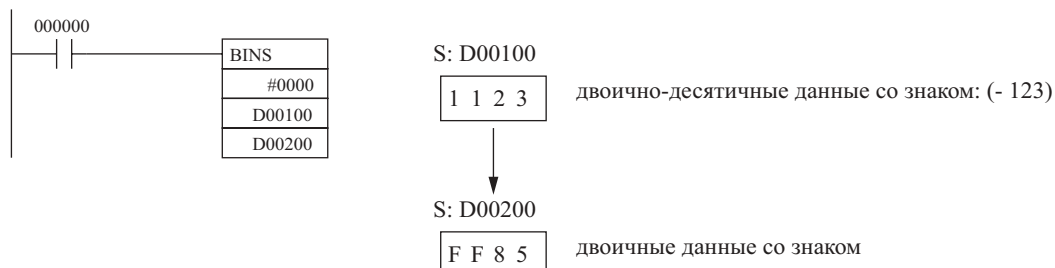
### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение C выходит за пределы допустимого диапазона значений от 0000 до 0003. Переводится в состояние ON, когда C=0002 и старшая цифра слова S равна от A до E. Переводится в состояние ON, когда C=0003 и старшая цифра слова S равна от B до E. Переводится в состояние ON, когда содержание S выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержание D равно 0000. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды бит 15 слова D равен 1. OFF в других случаях.

### Примеры

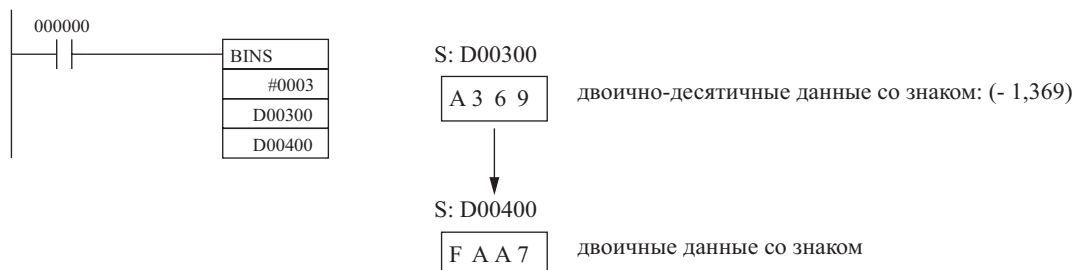
#### Формат 0 двоично-десятичных данных (C=#0000)

В следующем ниже примере, когда СЮ 000000 находится в состоянии ON, формат двоично-десятичных данных и диапазон данных, содержащихся в D00100, проверяется на соответствие формату, указанному в управляющем слове (0000). Если исходные данные верны, двоично-десятичные данные в D00100 преобразуются в двоичные данные со знаком и записываются в D00200.



**Формат 0 двоично-десятичных данных (C=#0003)**

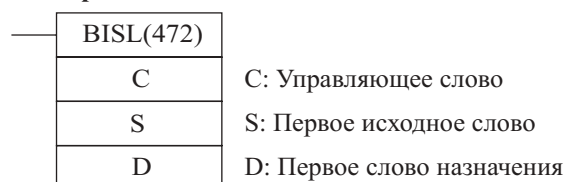
В следующем ниже примере, когда СЮ 000001 находится в состоянии ON, формат двоично-десятичных данных и диапазон данных, содержащихся в D00100, проверяется на соответствие формату, указанному в управляющем слове (0003). Если исходные данные верны, двоично-десятичные данные в D00300 преобразуются в двоичные данные со знаком и записываются в D00400.



### 3-12-15 Команда преобразования двойных двоично-десятичных данных со знаком в двойные двоичные данные со знаком DOUBLE SIGNED BCD-TO-BINARY: BISL(472)

**Назначение**

По команде BISL(472) осуществляется преобразование двойных двоично-десятичных данных со знаком в двойные двоичные данные со знаком.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	BISL(472)
	Выполняется один раз для дифференцирования вверх	@ BISL(472)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****C: Управляющее слово**

Указывает формат двоично-десятичных данных со знаком. Диапазон значений C- от 0000 до 0003.

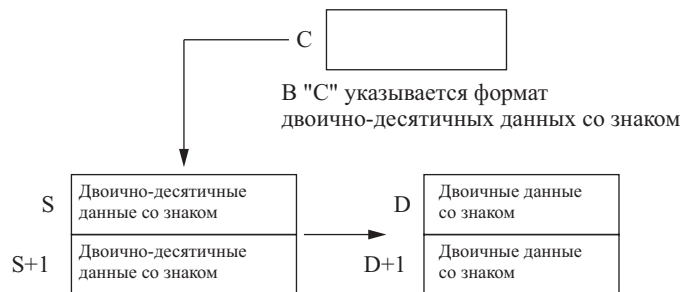
**Спецификации операндов**

Область	C	S	D
Область ввода/вывода (область СЮ)	СЮ 0000...СЮ 6143	СЮ 0000...СЮ 6142	
Рабочая область	W000...W511	W000...W510	
Область удержания бита (Holding bit area)	H000...H511	H000...H510	
Область вспомогательных битов	A000...A959	A000...A958	A448...A958
Область таймера	T0000...T4095	T0000...T4094	
Область счетчика	C0000...C4095	C0000...C4094	
Область DM	D00000...D32767	D00000...D32766	
Область EM, не содержащая банков	E00000...E32767	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		

Область	C	S	D
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#0003 (двоичные)	–	
Регистры данных	DR0...DR15	–	
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0... ,IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0... IR15 ,IR0+(++)... ,IR15+(++) ,(- -)IR0... ,(- -)IR15		

### Описание

По команде BISL(472) производится преобразование двойных двоично-десятичных данных со знаком, находящихся в словах S+1 и S, в двойные двоичные данные со знаком. Результат записывается в слова D+1 и D. Вначале в словах S+1 и S проверяется формат двоично-десятичных данных со знаком и диапазон значений данных на соответствие данным управляющего слова (C). Если данные в исходных словах верны, данные слов S+1 и S преобразуются в двоичные данные со знаком и записываются в слова D+1 и D. Если исходные данные некорректны, флаг ошибки переводится в состояние ON, и команда не выполняется.



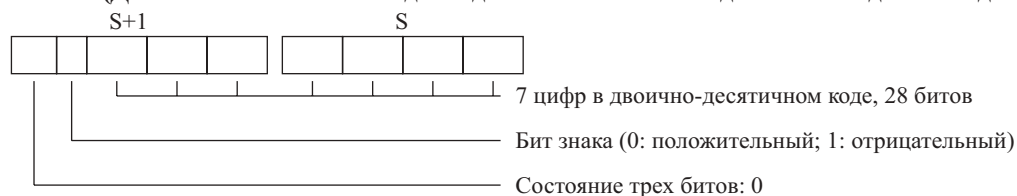
Когда преобразуемые данные являются отрицательным числом, они записываются в виде дополнения по модулю 2 к результирующему числу, и флаг отрицательного значения переводится в состояние ON. Для определения абсолютного значения отрицательного двоичного числа может использоваться команда NEG(160). Для ознакомления с деталями обратитесь к разделу 3-12-6 «Команда вычисления дополнения к двойному слову данных по модулю 2: NEGL(161)».

Значение исходных данных, равное «-0» будет обрабатываться как 0, и не приведет к определению ошибки. Кроме того, когда C=0000, значение битов 13...15 слова S+1 не проверяется.

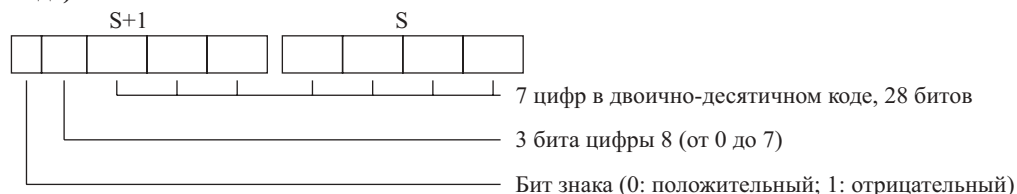
**Примечание:** Некоторые Специальные модули подают на выход двоично-десятичные данные со знаком. Обработка таких данных, связанная с вычислениями, обычно упрощается, если предварительно эти данные преобразовать в двоичные данные со знаком, используя команду BISI(472).

Управляющее слово определяет формат двоично-десятичных данных со знаком, как показано ниже.

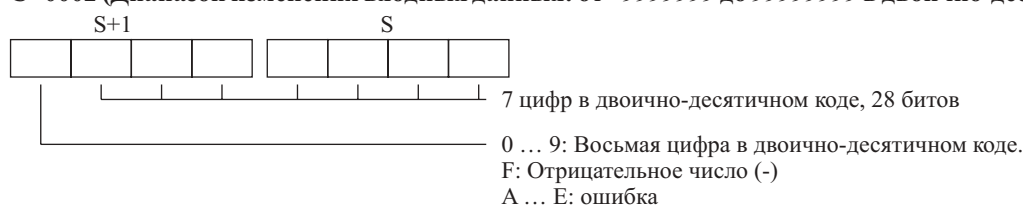
**C=0000 (Диапазон изменения входных данных: от -9999999 до 9999999 в двоично-десятичном коде)**



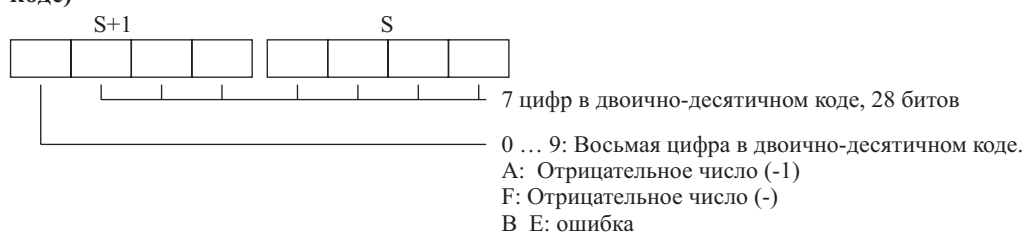
**C=0001 (Диапазон изменения входных данных: от -7999999 до 7999999 в двоично-десятичном коде)**



**C=0002 (Диапазон изменения входных данных: от -9999999 до 99999999 в двоично-десятичном коде)**



**C=0003 (Диапазон изменения входных данных: от -19999999 до 99999999 в двоично-десятичном коде)**



В следующей ниже таблице показаны возможные значения для каждого из двоично-десятичных форматов, и соответствующие двоичные числа со знаком.

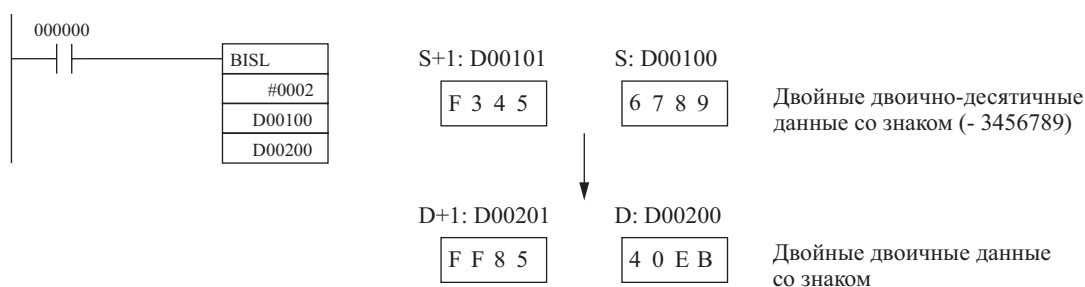
Установка	Двоично-десятичные значения со знаком	Двоичные значения со знаком
C=0000	-9999999...-1	FF676981...FFFFFFF
	0...9999999	00000000...0098967F
C=0001	-79999999...-1	FB3B4C01...FFFFFFF
	0...79999999	00000000...04CAB3FF
C=0002	-9999999...-1	FF676981...FFFFFFF
	0...9999999	00000000...05F5E0FF
C=0003	-19999999...-1	FECED301...FFFFFFF
	0...9999999	00000000...05F5E0FF

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение C выходит за пределы допустимого диапазона значений от 0000 до 0003. Переводится в состояние ON, когда C=0002 и старшая цифра слова S+1 находится в пределах от A до E. Переводится в состояние ON, когда C=0003 и старшая цифра слова S+1 находится в пределах от B до E. Переводится в состояние ON, когда содержание S+1 и S выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержание D+1 равно 00000000. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды бит 15 слова D+1 равен 1. OFF в других случаях.

### Пример

В следующем ниже примере, когда СЮ 000000 находится в состоянии ON, формат двойных двоично-десятичных данных и диапазон данных, содержащихся в D00101 и D00100, проверяется на соответствие формату, указанному в управляющем слове (0002). Если исходные данные верны, двоично-десятичные данные в D00101 и D00100 преобразуются в двоичные данные со знаком и записываются в D00201 и D00200.

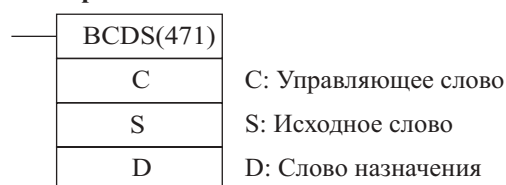


### 3-12-16 Команда преобразования двоичных данных со знаком в двоично-десятичные данные со знаком SIGNED BINARY-TO-BCD: BCDS(471)

#### Назначение

По команде BCDS(471) осуществляется преобразование одного слова двоичных данных со знаком в одно слово двоично-десятичных данных со знаком.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	BCDS(471)
	Выполняется один раз для дифференцирования вверх	@ BCDS(471)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

##### C: Управляющее слово

Указывает формат двоично-десятичных данных со знаком. Диапазон значений C - от 0000 до 0003.

##### S: Исходное слово

Содержит двоичные данные со знаком, подлежащие преобразованию. Содержание S должно находиться в пределах диапазона, определяемого двоично-десятичным форматом, указанным в слове C.

Установка	Допустимые значения S
C=0000	FC19...FFFF или 0000...03E7
C=0001	E0C1...FFFF или 0000...1F3F
C=0002	FC19...FFFF или 0000...270F
C=003	F831...FFFF или 0000...270F

##### D: Слово назначения

Содержит данные, преобразованные в двоично-десятичное значение со знаком. Пояснения к форматам двоично-десятичных данных изложены ниже.

#### Спецификации операндов

Область	C	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	



Область	C	S	D
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#0003 (двоичные)	–	
Регистры данных	DR0...DR15		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0... IR15 ,IR0+(++)...IR15(++) ,(- -)IR0...(- -)IR15		

### Описание

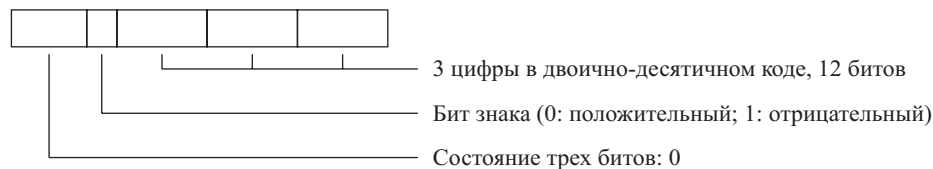
По команде BCDS(471) осуществляется преобразование двоичных данных со знаком в двоично-десятичные данные со знаком. Вначале двоично-десятичные данные со знаком в слове S проверяются на соответствие формату, указанному в управляющем слове (C). Если данные в исходном слове верны, двоичные данные со знаком в слове S преобразуются в двоично-десятичные данные со знаком и записываются в слово D. Если исходные данные некорректны, флаг ошибки переводится в состояние ON, и команда не выполняется.



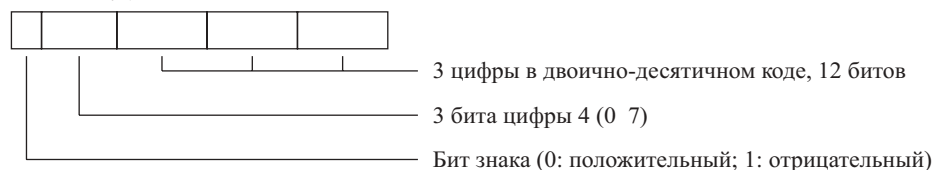
- Примечание:**
1. Значение исходных данных, равное «-0» будет обрабатываться как 0, и не приведет к определению ошибки.
  2. Некоторые модели Специальных модулей требуют подачи на вход двоично-десятичных данных со знаком. В этом случае для преобразования двоичных данных и подачи их на вход Модуля используйте команду BCDS(471).

Управляющее слово определяет используемый для вывода результата формат двоично-десятичных данных со знаком, как показано ниже.

**C=0000 (Диапазон изменения выходных данных: от -999 до 999 в двоично-десятичном коде)**



**C=0001 (Диапазон изменения выходных данных: от -7999 до 7999 в двоично-десятичном коде)**



**C=0002 (Диапазон изменения выходных данных: от –999 до 9999 в двоично-десятичном коде)**



3 цифры в двоично-десятичном коде, 12 битов

0 ... 9: Четвертая цифра в двоично-десятичном коде.

F: Отрицательное число (-)

**C=0003 (Диапазон изменения выходных данных: от –1999 до 9999 в двоично-десятичном коде)**



3 цифры в двоично-десятичном коде, 12 битов

0 ... 9: Четвертая цифра в двоично-десятичном коде.

A: Отрицательное число (-1)

F: Отрицательное число (-)

В следующей ниже таблице показаны возможные двоичные значения со знаком для каждого из двоично-десятичных форматов. Если данные исходного слова выходят за допускаемые пределы диапазона, определяется ошибка.

Установка	Двоичные значения со знаком	Двоично-десятичные значения со знаком
C=0000	FC19...FFFF и 0000...03E7	-999...-1 и 0...999
C=0001	E0C1...FFFF и 0000...1F3F	-7999...-1 и 0...7999
C=0002	FC19...FFFF и 0000...270F	-999...-1 и 0...9999
C=003	F831...FFFF и 0000...270F	-1999...-1 и 0...9999

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение C выходит за пределы допускаемого диапазона значений от 0000 до 0003. Переводится в состояние ON, когда C=0000 и исходные данные находятся за пределами диапазона (FC19 FFFF и 0000 03E7). Переводится в состояние ON, когда C=0001 и исходные данные находятся за пределами диапазона (E0C1 FFFF и 0000 1F3F). Переводится в состояние ON, когда C=0002 и исходные данные находятся за пределами диапазона (FC19 FFFF и 0000 270F). Переводится в состояние ON, когда C=0003 и исходные данные находятся за пределами диапазона (F831 FFFF и 0000 270F). OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержание D равно 0000. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда C=0000 или C=0001 и после выполнения команды бит знака равен 1. Переводится в состояние ON, когда C=0002 и старший бит результирующего слова равен F. Переводится в состояние ON, когда C=0003 и старший бит результирующего слова равен A или F. OFF в других случаях.

### 3-12-17 Команда преобразования двойных двоичных данных со знаком в двойные двоично-десятичные данные со знаком DOUBLE SIGNED BINARY-TO-BCD: BDSL(473)

#### Назначение

По команде BDSL(473) осуществляется преобразование двойного слова двоичных данных со знаком в двойное слово двоично-десятичных данных со знаком.

## Символ релейно-контактной схемы

BDSL(473)	
C	C: Управляющее слово
S	S: Первое исходное слово
D	D: Первое слово назначения

## Модификации

Модификации	Выполняется в каждом цикле для состояния ON	BDSL(473)
	Выполняется один раз для дифференцирования вверх	@ BDSL(473)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

## Операнды

**C: Управляющее слово**

Указывает формат двоично-десятичных данных со знаком. Диапазон значений C - от 0000 до 0003.

**S: Первое исходное слово**

Исходные слова S+1 и S содержат двойные двоичные данные со знаком, подлежащие преобразованию. Содержание исходных слов должно находиться в пределах диапазона, определяемого двоично-десятичным форматом, указанным в слове C.

Установка	Допустимые значения S
C=0000	FF67 6981 FFFFFFFF или 00000000 0098967F
C=0001	FB3B 4C01 FFFFFFFF или 00000000 04C4B3FF
C=0002	FF676981 FFFFFFFF или 00000000 05F5 E0FF
C=0003	FECED301 FFFFFFFF или 00000000 05F5 E0FF

**D: Слово назначения**

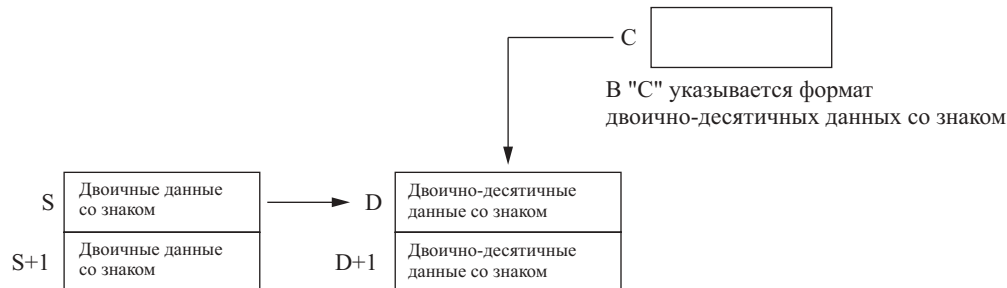
Слова D+1 и D содержат двойные данные, преобразованные в двоично-десятичное значение со знаком. Пояснения к форматам двоично-десятичных данных изложены ниже.

## Спецификации операндов

Область	C	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6142	
Рабочая область	W000...W511	W000...W510	
Область удержания бита (Holding bit area)	H000...H511	H000...H510	
Область вспомогательных битов	A000...A959	A000...A958	A448...A959
Область таймера	T0000...T4095	T0000...T4094	
Область счетчика	C0000...C4095	C0000...C4094	
Область DM	D00000...D32767	D00000...D32766	
Область EM, не содержащая банков	E00000...E32767	E00000...E32766	
20Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#0003 (двоичные)	—	
Регистры данных	DR0...DR15	—	
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047,IR0 -2048... +2047,IR15 DR0...DR15, IR0... IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15		

**Описание**

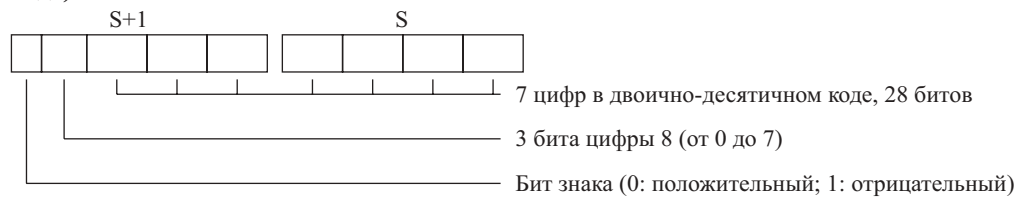
По команде BDSL(473) осуществляется преобразование двойных двоичных данных со знаком в двойные двоично-десятичные данные со знаком. Вначале двоично-десятичные данные со знаком в словах S+1 и S проверяются на соответствие формату, указанному в управляющем слове (C). Если данные в исходном слове верны, двоичные данные со знаком в словах S+1 и S преобразуются в двоично-десятичные данные со знаком и записываются в слова D+1 и D. Если исходные данные некорректны, флаг ошибки переводится в состояние ON, и команда не выполняется.



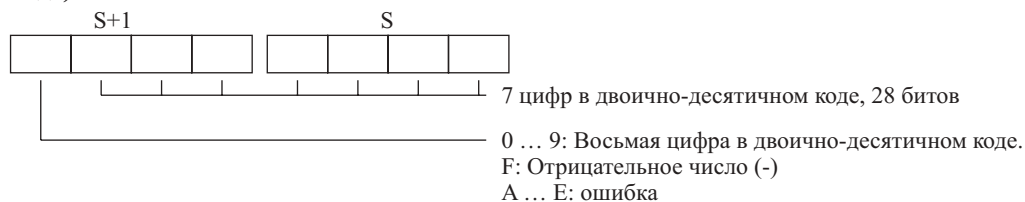
- Примечание:**
1. Значение исходных данных, равное «-0» будет обрабатываться как 0, и не приведет к определению ошибки.
  2. Некоторые модели Специальных модулей требуют подачи на вход двоично-десятичных данных со знаком. В этом случае для преобразования двоичных данных и подачи их на вход модуля используйте команду BDSL(473).

Управляющее слово определяет используемый для вывода результата формат двоично-десятичных данных со знаком, как показано ниже.

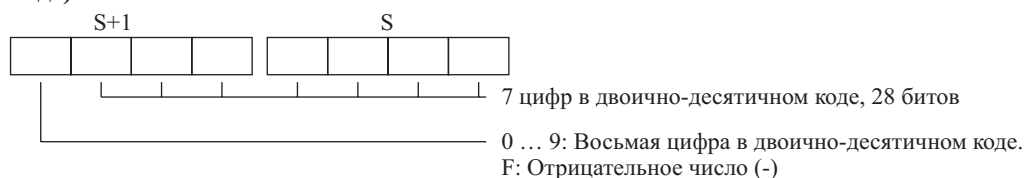
**C=0000 (Диапазон изменения выходных данных: от -9999999 до 9999999 в двоично-десятичном коде)**



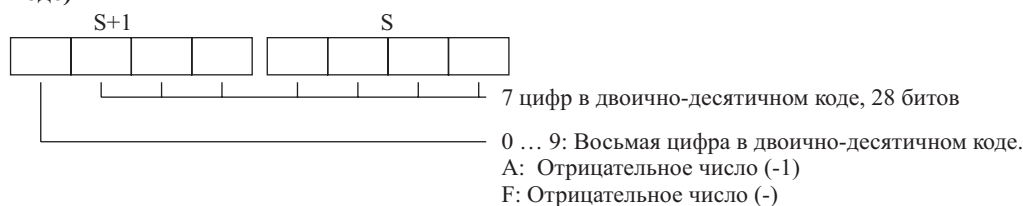
**C=0001 (Диапазон изменения выходных данных: от -79999999 до 79999999 в двоично-десятичном коде)**



**C=0002 (Диапазон изменения выходных данных: от -9999999 до 99999999 в двоично-десятичном коде)**



**C=0003 (Диапазон изменения выходных данных: от -19999999 до 99999999 в двоично-десятичном коде)**



В следующей ниже таблице показаны возможные двойные двоичные числа со знаком для каждого из двоично-десятичных форматов. Если данные исходного слова выходят за допускаемые пределы диапазона, определяется ошибка.

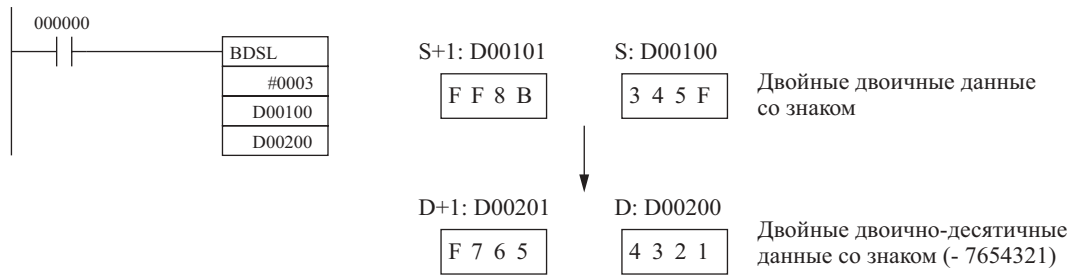
Установка	Двоичные значения со знаком	Двоично-десятичные значения со знаком
C=0000	FF67 6981...FFFFFFFF	-9999999...-1
	00000000...0098967F	0...9999999
C=0001	FB3B 4C01...FFFFFFFF	-79999999...-1
	00000000...04C4B3FF	0...79999999
C=0002	FF676981...FFFFFFFF	-9999999...-1
	00000000...05F5 E0FF	0...99999999
C=003	FECED301...FFFFFFFF	-1999...-1
	00000000...05F5 E0FF	0...99999999

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение C выходит за пределы допускаемого диапазона значений от 0000 до 0003. Переводится в состояние ON, когда C=0000 и исходные данные находятся за пределами диапазона (FF67 6981 FFFFFFFF или 00000000 0098967F). Переводится в состояние ON, когда C=0001 и исходные данные находятся за пределами диапазона (FB3B 4C01 FFFFFFFF или 00000000 04C4B3FF). Переводится в состояние ON, когда C=0002 и исходные данные находятся за пределами диапазона (FF676981 FFFFFFFF или 00000000 05F5 E0FF). Переводится в состояние ON, когда C=0003 и исходные данные находятся за пределами диапазона (FECED301 FFFFFFFF или 00000000 05F5 E0FF). OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержание D равно 0000. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда C=0000 или C=0001 и после выполнения команды бит знака равен 1. Переводится в состояние ON, когда C=0002 и старший бит результирующего слова равен F. Переводится в состояние ON, когда C=0003 и старший бит результирующего слова равен A или F. OFF в других случаях.

#### Пример

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, двойные двоичные данные в D00101 и D00100, проверяется на соответствие формату, указанному в управляющем слове (0003). Если исходные данные верны, двойные двоичные данные в D00101 и D00100 преобразуются в двойные двоично-десятичные данные со знаком и записываются в D00201 и D00200.



### 3-13 Логические команды

В настоящем разделе приводится описание команд, используемых для выполнения логических операций с данными в виде слов.

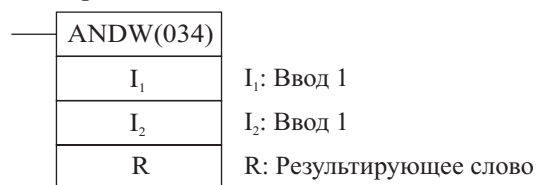
Команда	Мнемоническое изображение	Код функции	Страница
LOGICAL AND	ANDW	034	384
DOUBLE LOGICAL AND	ANDL	610	385
LOGICAL OR	ORW	035	387
DOUBLE LOGICAL OR	ORWL	611	388
EXCLUSIVE OR	XORW	036	390
DOUBLE EXCLUSIVE OR	XORL	612	391
EXCLUSIVE NOR	XNRW	037	393
DOUBLE EXCLUSIVE NOR	XNRL	613	395
COMPLEMENT	COM	029	397
DOUBLE COMPLEMENT	COML	614	398

#### 3-13-1 Команда выполнения логической операции «И» LOGICAL AND: ANDW(034)

##### Назначение

По команде ANDW(034) осуществляется выполнение логической операции «И» с соответствующими битами простых слов данных и/или константами.

##### Символ релейно-контактной схемы



##### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	ANDW(034)
	Выполняется один раз для дифференцирования вверх	@ ANDW(034)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

##### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

##### Спецификации операндов

Область	I <sub>1</sub>	I <sub>2</sub>	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)		

Область	I <sub>1</sub>	I <sub>2</sub>	R
Константы	#0000...#FFFF (двоичные)	—	
Регистры данных	DR0...DR15		
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15(++) ,(- -)IR0...,(- -)IR15		

**Описание**

По команде ANDW(034) осуществляется выполнение логической операции «И» с данными, указанными в I<sub>1</sub> и I<sub>2</sub>. Результат выводится в слово R.

- Логическая операция «И» выполняется последовательно с соответствующими битами в словах I<sub>1</sub> и I<sub>2</sub>.
- Когда содержанием соответствующих битов как в I<sub>1</sub>, так и в I<sub>2</sub> является 1 или один из них равен 0, в соответствующий бит слова R записывается 0.

$I_1, I_2 \rightarrow R$

I <sub>1</sub>	I <sub>2</sub>	R
1	1	1
1	0	0
0	1	0
0	0	0

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержание R равно нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующего слова R равен 1. OFF в других случаях.

**Меры предосторожности**

После выполнения команды ANDW(034) флаг ошибки переводится в состояние OFF.

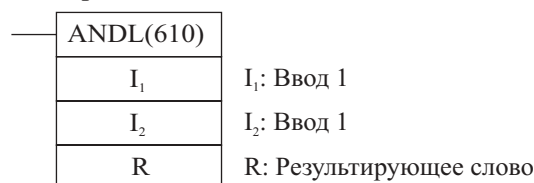
Когда в результате выполнения команды ANDW(034) содержание результирующего слова R равно 0000 (шестн.) флаг равенства переводится в состояние ON.

Когда в результате выполнения команды ANDW(034) старший бит результирующего слова R равен 1, флаг отрицательного значения переводится в состояние ON.

### 3-13-2 Команда выполнения логической операции «И» с двойными данными DOUBLE LOGICAL AND: ANDL(610)

**Назначение**

По команде ANDL(610) осуществляется выполнение логической операции «И» с соответствующими битами двойных слов данных и/или константами.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	ANDL(610)
	Выполняется один раз для дифференцирования вверх	@ ANDL(610)
	Выполняется один раз для дифференцирования вниз	Не поддерживается



Указание немедленной ре-генерации.	Не поддерживается
------------------------------------	-------------------

### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

### Спецификации операндов

Область	I <sub>1</sub>	I <sub>2</sub>	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142		
Рабочая область	W000...W510		
Область удержания бита (Holding bit area)	H000...H510		
Область вспомогательных битов	A000...A958	A448...A958	
Область таймера	T0000...T4094		
Область счетчика	C0000...C4094		
Область DM	D00000...D32766		
Область EM, не содержащая банков	E00000...E32766		
Область EM, содержащая банки	En_00000...En_32766 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)		
Константы	#00000000...#FFFF FFFF (двоичные)	–	
Регистры данных	–		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15		

### Описание

По команде ANDL(610) осуществляется выполнение логической операции «И» с данными, указанными в I<sub>1</sub>, I<sub>1</sub>+1, и I<sub>2</sub>, I<sub>2</sub>+1. Результат выводится в слова R и R+1.

(I<sub>1</sub>, I<sub>1</sub>+1), (I<sub>2</sub>, I<sub>2</sub>+1) → (R, R+1)

I <sub>1</sub> , I <sub>1</sub> +1	I <sub>2</sub> , I <sub>2</sub> +1	R, R+1
1	1	1
1	0	0
0	1	0
0	0	0

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующего слова R+1 равен 1. OFF в других случаях.

### Меры предосторожности

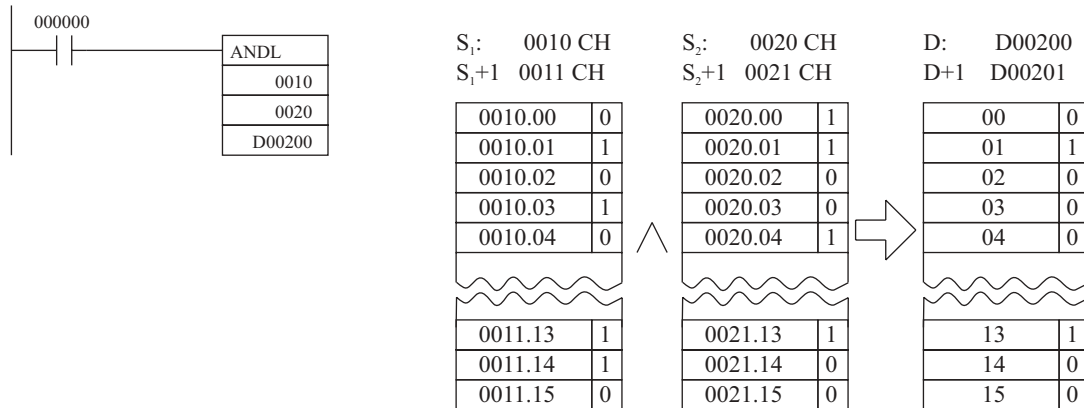
После выполнения команды ANDL(610) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения команды ANDL(610) содержание результирующих слов R, R+1 равно 00000000 (шестн.) флаг равенства переводится в состояние ON.

Когда в результате выполнения команды ANDL(610) старший бит результирующего слова R+1 равен 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

Когда условие выполнения СЮ 00000000 находится в состоянии ON, производится выполнение логической операции «И» с соответствующими битами в словах СЮ 0011, СЮ 0010 и СЮ 0021, СЮ 0020. Результат записывается в соответствующие биты в словах D00201 и D00200.



Примечание: Вертикальная стрелка обозначает логическую операцию "И"

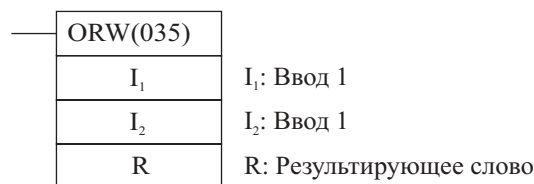
*Примечание:* Вертикальная стрелка обозначает логическую операцию «И».

**3-13-3 Команда выполнения логической операции «ИЛИ» LOGICAL OR: ORW(035)**

**Назначение**

По команде ORW(035) осуществляется выполнение логической операции «ИЛИ» с соответствующими битами простых слов данных и/или константами.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	ORW(035)
	Выполняется один раз для дифференцирования вверх	@ ORW(035)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	I <sub>1</sub>	I <sub>2</sub>	R
Область ввода/вывода (область СЮ)	СЮ 0000...СЮ 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0... C)		

Область	I <sub>1</sub>	I <sub>2</sub>	R
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)		
Константы	#0000...#FFFF (двоичные)	–	
Регистры данных	DR0...DR15		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(++ ,(- -)IR0...,(- -)IR15		

**Описание**

По команде ORW(035) осуществляется выполнение логической операции «ИЛИ» с данными, указанными в I<sub>1</sub> и I<sub>2</sub>. Результат выводится в слово R.

Логическая операция «ИЛИ» выполняется последовательно с соответствующими битами в словах I<sub>1</sub> и I<sub>2</sub>.

Когда один из битов в I<sub>1</sub> и в I<sub>2</sub> равен 1 или оба бита равны 0, в соответствующий бит слова R записывается 1.

$$I_1 + I_2 \rightarrow R$$

I <sub>1</sub>	I <sub>2</sub>	R
1	1	1
1	0	1
0	1	1
0	0	0

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержание R равно нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующего слова R равен 1. OFF в других случаях.

**Меры предосторожности**

После выполнения команды ORW(035) флаг ошибки переводится в состояние OFF.

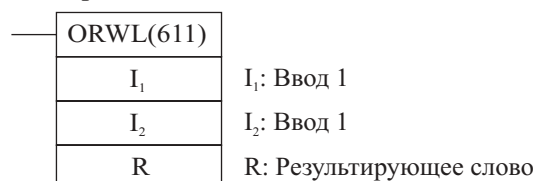
Когда в результате выполнения команды ORW(035) содержание результирующего слова R равно 0000 (шестн.) флаг равенства переводится в состояние ON.

Когда в результате выполнения команды ORW(034) старший бит результирующего слова R равен 1, флаг отрицательного значения переводится в состояние ON.

### 3-13-4 Команда выполнения логической операции «ИЛИ» с двойными данными DOUBLE LOGICAL OR: ORWL(611)

**Назначение**

По команде ORWL(611) осуществляется выполнение логической операции «ИЛИ» с соответствующими битами двойных слов данных и/или константами.

**Символ релейно-контактной схемы**

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	ORWL(611)
	Выполняется один раз для дифференцирования вверх	@ ORWL(611)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	I <sub>1</sub>	I <sub>2</sub>	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142		
Рабочая область	W000...W510		
Область удержания бита (Holding bit area)	H000...H510		
Область вспомогательных битов	A000...A958	A448...A958	
Область таймера	T0000...T4094		
Область счетчика	C0000...C4094		
Область DM	D00000...D32766		
Область EM, не содержащая банков	E00000...E32766		
Область EM, содержащая банки	En_00000...En_32766 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)		
Константы	#00000000...#FFFF FFFF (двоичные)	—	
Регистры данных	—		
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15		

**Описание**

По команде ORWL(611) осуществляется выполнение логической операции «ИЛИ» с данными в виде двойного слова, указанными в I<sub>1</sub>, I<sub>1</sub>+1, и I<sub>2</sub>, I<sub>2</sub>+1. Результат выводится в слова R и R+1.

Когда любой из битов в I<sub>1</sub>, I<sub>1</sub>+1 и в I<sub>2</sub>, I<sub>2</sub>+1 равен 1, в соответствующий бит слова R+1 записывается 1. Когда оба бита равны 0, в соответствующий бит слова R+1 записывается 0.

$$(I_1, I_1 + 1) + (I_2, I_2 + 1) \rightarrow (R, R + 1)$$

I <sub>1</sub> , I <sub>1</sub> +1	I <sub>2</sub> , I <sub>2</sub> +1	R, R+1
1	1	1
1	0	1
0	1	1
0	0	0

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующего слова R+1 равен 1. OFF в других случаях.

**Меры предосторожности**

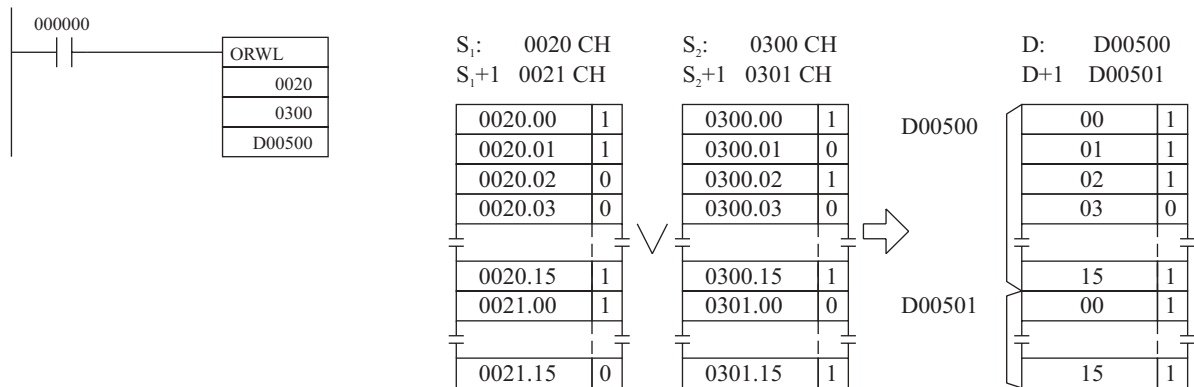
После выполнения команды ORWL(611) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения команды ORWL(611) содержание результирующих слов R, R+1 равно 00000000 (шестн.) флаг равенства переводится в состояние ON.

Когда в результате выполнения команды ORWL(611) старший бит результирующего слова R+1 равен 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

Когда условие выполнения CIO 00000000 находится в состоянии ON, производится выполнение логической операции «ИЛИ» с соответствующими битами в словах CIO 0021, CIO 0020 и CIO 0301, CIO 0300. Результат записывается в соответствующие биты в словах D00501 и D00500.



Вертикальная стрелка обозначает логическую операцию "ИЛИ"

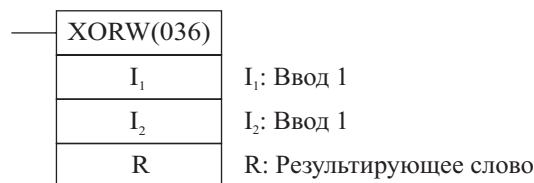
*Примечание:* Вертикальная стрелка обозначает логическую операцию «ИЛИ» (∨).

**3-13-5 Команда выполнения логической операции «Исключающее ИЛИ» EXCLUSIVE OR: XORW(036)**

**Назначение**

По команде XORW(036) осуществляется выполнение логической операции «Исключающее ИЛИ» с соответствующими битами простых слов данных и/или константами.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	XORW(036)
	Выполняется один раз для дифференцирования вверх	@ XORW(036)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	I <sub>1</sub>	I <sub>2</sub>	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		

Область	I <sub>1</sub>	I <sub>2</sub>	R
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)		
Константы	#0000...#FFFF (двоичные)	–	
Регистры данных	DR0...DR15		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15		

**Описание**

По команде XORW(036) осуществляется выполнение логической операции «Исключающее ИЛИ» с данными, указанными в I<sub>1</sub> и I<sub>2</sub>. Результат выводится в слово R.

Логическая операция «Исключающее ИЛИ» выполняется последовательно с соответствующими битами в словах I<sub>1</sub> и I<sub>2</sub>.

Когда содержание битов в I<sub>1</sub> и в I<sub>2</sub> различается, в соответствующий бит слова R записывается 1, а когда содержание битов не отличается друг от друга, в соответствующий бит слова R записывается 0.

$$I_1, I_2 + I_1, I_2 \rightarrow R$$

I <sub>1</sub>	I <sub>2</sub>	R
1	1	0
1	0	1
0	1	1
0	0	0

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержание R равно нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующего слова R равен 1. OFF в других случаях.

**Меры предосторожности**

После выполнения команды XORW(036) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения команды XORW(036) содержание результирующего слова R равно 0000 (шестн.) флаг равенства переводится в состояние ON.

Когда в результате выполнения команды XORW(036) старший бит результирующего слова R равен 1, флаг отрицательного значения переводится в состояние ON.

### 3-13-6 Команда выполнения логической операции «Исключающее ИЛИ» с двойными данными DOUBLE EXCLUSIVE OR: XORL(612)

**Назначение**

По команде XORL(612) осуществляется выполнение логической операции «Исключающее ИЛИ» с соответствующими битами двойных слов данных и/или константами.

**Символ релейно-контактной схемы**

XORL(612)	
I <sub>1</sub>	I <sub>1</sub> : Ввод 1
I <sub>2</sub>	I <sub>2</sub> : Ввод 1
R	R: Результирующее слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	XORL(612)
	Выполняется один раз для дифференцирования вверх	@ XORL(612)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	I <sub>1</sub>	I <sub>2</sub>	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142		
Рабочая область	W000...W510		
Область удержания бита (Holding bit area)	H000...H510		
Область вспомогательных битов	A000...A958	A448...A958	
Область таймера	T0000...T4094		
Область счетчика	C0000...C4094		
Область DM	D00000...D32766		
Область EM, не содержащая банков	E00000...E32766		
Область EM, содержащая банки	En_00000...En_32766 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)		
Константы	#00000000...#FFFF FFFF (двоичные)	-	
Регистры данных	-		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047,IR0 -2048... +2047,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(-)IR0...,(-)IR15		

**Описание**

По команде XORL(612) осуществляется выполнение логической операции «Исключающее ИЛИ» с данными в виде двойного слова, указанными в I<sub>1</sub>, I<sub>1</sub>+1, и I<sub>2</sub>, I<sub>2</sub>+1. Результат выводится в слова R и R+1. Когда содержание соответствующих битов в I<sub>1</sub>, I<sub>1</sub>+1 и в I<sub>2</sub>, I<sub>2</sub>+1 различается, в соответствующий бит слов R, R+1 записывается 1. Когда оба соответствующих бита равны, в соответствующий бит слов R, R+1 записывается 0.

$$(I_1, I_1 + 1), (I_2, I_2 + 1) + (\overline{I_1}, \overline{I_1 + 1}), (\overline{I_2}, \overline{I_2 + 1}) \rightarrow (R, R + 1)$$

I <sub>1</sub> , I <sub>1</sub> +1	I <sub>2</sub> , I <sub>2</sub> +1	R, R+1
1	1	0
1	0	1
0	1	1
0	0	0

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующего слова R+1 равен 1. OFF в других случаях.

**Меры предосторожности**

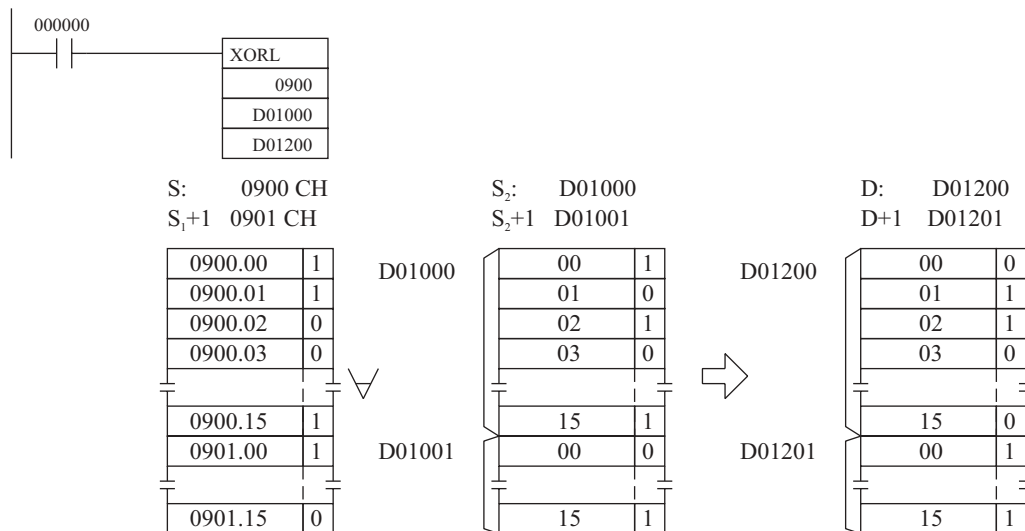
После выполнения команды XORL(612) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения команды XORL(612) содержание результирующих слов R, R+1 равно 00000000 (шестн.) флаг равенства переводится в состояние ON.

Когда в результате выполнения команды XORL(612) старший бит результирующего слова R+1 равен 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

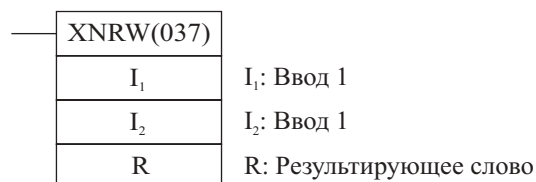
Когда условие выполнения CIO 00000000 находится в состоянии ON, производится выполнение логической операции «Исключающее ИЛИ» с соответствующими битами в словах CIO 0901, CIO 0900 и D01001, D01000. Результат записывается в соответствующие биты в словах D01201 и D01200.



Примечание: Символ обозначает логическую операцию "Исключающее ИЛИ"

**3-13-7 Команда выполнения логической операции «Исключающее ИЛИ-НЕ» EXCLUSIVE NOR: XNRW(037)****Назначение**

По команде XNRW(037) осуществляется выполнение логической операции «Исключающее ИЛИ-НЕ» с соответствующими битами простых слов данных и/или константами.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	XNRW(037)
	Выполняется один раз для дифференцирования вверх	@ XNRW(037)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	I <sub>1</sub>	I <sub>2</sub>	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)		
Константы	#0000...#FFFF (двоичные)	–	
Регистры данных	DR0...DR15		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15		

**Описание**

По команде XNRW(037) осуществляется выполнение логической операции «Исключающее ИЛИ-НЕ» с данными, указанными в I<sub>1</sub> и I<sub>2</sub>. Результат выводится в слово R.

Логическая операция «Исключающее ИЛИ-НЕ» выполняется последовательно с соответствующими битами в словах I<sub>1</sub> и I<sub>2</sub>.

Когда содержание соответствующих битов в I<sub>1</sub> и в I<sub>2</sub> различается, в соответствующий бит слова R записывается 0, а когда содержание битов не отличается друг от друга, в соответствующий бит слова R записывается 1.

$I_1, I_2 + I_1, I_2 \rightarrow R$

I <sub>1</sub>	I <sub>2</sub>	R
1	1	1
1	0	0
0	1	0
0	0	1

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержание R равно нулю. OFF в других случаях.

Наименование	Обозначение	Функционирование
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующего слова R равен 1. OFF в других случаях.

**Меры предосторожности**

После выполнения команды XNRW(037) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения команды XNRW(037) содержание результирующего слова R равно 0000 (шестн.) флаг равенства переводится в состояние ON.

Когда в результате выполнения команды старший бит результирующего слова R равен 1, флаг отрицательного значения переводится в состояние ON.

**3-13-8 Команда выполнения логической операции «Исключающее ИЛИ-НЕ» с двойными данными DOUBLE EXCLUSIVE NOR: XNRL(613)****Назначение**

По команде XNRL(613) осуществляется выполнение логической операции «Исключающее ИЛИ-НЕ» с соответствующими битами двойных слов данных и/или константами.

**Символ релейно-контактной схемы**

XNRL(613)	
I <sub>1</sub>	I <sub>1</sub> : Ввод 1
I <sub>2</sub>	I <sub>2</sub> : Ввод 1
R	R: Результирующее слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	XNRL(613)
	Выполняется один раз для дифференцирования вверх	@ XNRL(613)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	I <sub>1</sub>	I <sub>2</sub>	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142		
Рабочая область	W000...W510		
Область удержания бита (Holding bit area)	H000...H510		
Область вспомогательных битов	A000...A958	A448...A958	
Область таймера	T0000...T4094		
Область счетчика	C0000...C4094		
Область DM	D00000...D32766		
Область EM, не содержащая банков	E00000...E32766		
Область EM, содержащая банки	En_00000...En_32766 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)		
Константы	#00000000...#FFFF FFFF (двоичные)	—	
Регистры данных	—		
Индексные регистры	—		

Область	I <sub>1</sub>	I <sub>2</sub>	R
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15		

**Описание**

По команде XNRL(613) осуществляется выполнение логической операции «Исключающее ИЛИ-НЕ» с данными в виде двойного слова, указанными в I<sub>1</sub>, I<sub>1</sub>+1, и I<sub>2</sub>, I<sub>2</sub>+1. Результат выводится в слова R и R+1. Когда содержание соответствующих битов в I<sub>1</sub>, I<sub>1</sub>+1 и в I<sub>2</sub>, I<sub>2</sub>+1 различается, в соответствующий бит слов R, R+1 записывается 0. Когда оба соответствующих бита равны, в соответствующий бит слов R, R+1 записывается 1.

$$(I_1, I_1 + 1), (I_2, I_2 + 1) + (\overline{I_1, I_1 + 1}, \overline{I_2, I_2 + 1}) \rightarrow (R, R + 1)$$

I <sub>1</sub> , I <sub>1</sub> +1	I <sub>2</sub> , I <sub>2</sub> +1	R, R+1
1	1	0
1	0	1
0	1	1
0	0	0

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующего слова R+1 равен 1. OFF в других случаях.

**Меры предосторожности**

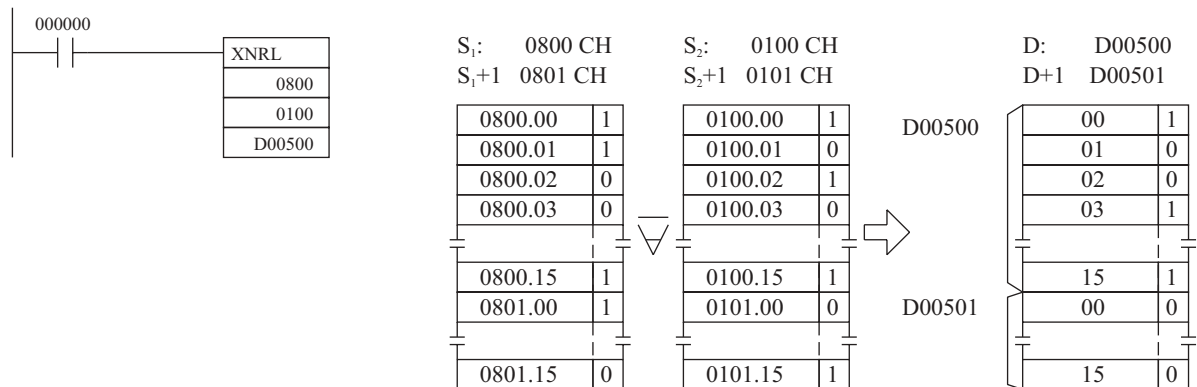
После выполнения команды XNRL(613) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения команды XNRL(613) содержание результирующих слов R, R+1 равно 00000000 (шестн.) флаг равенства переводится в состояние ON.

Когда в результате выполнения команды старший бит результирующего слова R+1 равен 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

Когда условие выполнения СЮ 00000000 находится в состоянии ON, производится выполнение логической операции «Исключающее ИЛИ-НЕ» с соответствующими битами в словах СЮ 0801, СЮ 0800 и СЮ 0101, СЮ 0100. Результат записывается в соответствующие биты в словах D00501 и D00500.



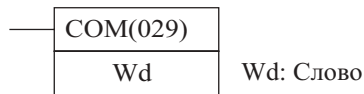
Примечание: Символ обозначает логическую операцию "Исключающее ИЛИ-НЕ"

**Примечание:** Символ обозначает логическую операцию «Исключающее ИЛИ-НЕ».

## 3-13-9 Команда вычисления дополнения COMPLEMENT: COM(029)

**Назначение**

По команде COM(029) осуществляется инвертирование состояния всех битов слова Wd.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	COM(029)
	Выполняется один раз для дифференцирования вверх	@ COM(029)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143
Рабочая область	W000...W511
Область удержания бита (Holding bit area)	H000...H511
Область вспомогательных битов	A448...A959
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область DM	D00000...D32767
Область EM, не содержащая банков	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0... C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)
Константы	—
Регистры данных	DR0...DR15
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-)IR15

**Описание**

По команде COM(029) производится инвертирование состояния всех битов слова Wd.

$Wd \rightarrow \bar{Wd}$ :  $1 \rightarrow 0$  и  $0 \rightarrow 1$

**Примечание:** При использовании команды COM(029) не забудьте, что состояние каждого из битов будет изменяться в каждом из циклов, в котором условие выполнения находится в состоянии ON.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержание R равно нулю. OFF в других случаях.

Наименование	Обозначение	Функционирование
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующего слова R равен 1. OFF в других случаях.

**Меры предосторожности**

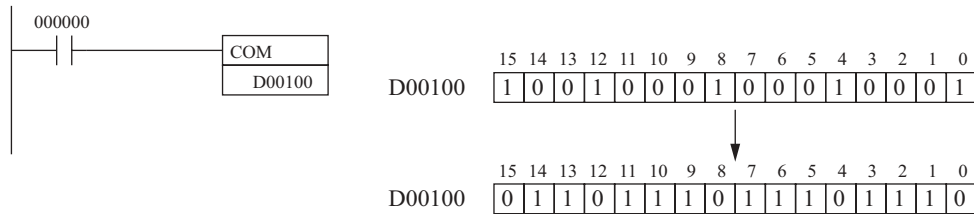
После выполнения команды COM(029) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения команды COM(029) содержание результирующего слова R равно 0000 (шестн.) флаг равенства переводится в состояние ON.

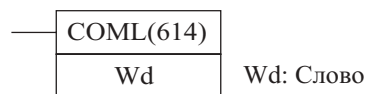
Когда в результате выполнения команды старший бит результирующего слова R равен 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

Когда в следующем ниже примере CIO 000000 находится в состоянии ON, состояние каждого из битов слова D00100 будет изменяться на противоположное.

**3-13-10 Команда вычисления дополнения к двойному слову DOUBLE COMPLEMENT: COML(614)****Назначение**

По команде COML(614) осуществляется инвертирование состояния всех битов слов Wd и Wd+1.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	COML(614)
	Выполняется один раз для дифференцирования вверх	@ COML(614)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации.	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Wd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142
Рабочая область	W000...W510
Область удержания бита (Holding bit area)	H000...H510
Область вспомогательных битов	A448...A958
Область таймера	T0000...T4094
Область счетчика	C0000...C4094
Область DM	D00000...D32766
Область EM, не содержащая банков	E00000...E32766
Область EM, содержащая банки	En_00000...En_32766 (n = 0... C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)

Область	Wd
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)
Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048... +2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...IR15(++) ,(- -)IR0...(- -)IR15

**Описание**

По команде COM(029) производится инвертирование состояния всех битов в словах Wd и Wd+1.  
 $(Wd + 1, Wd) \rightarrow (Wd + 1, Wd)$

**Примечание:** При использовании команды COML не забудьте, что состояние каждого из битов будет изменяться в каждом из циклов, в котором условие выполнения находится в состоянии ON.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержание R равно нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда после выполнения команды старший бит результирующего слова R равен 1. OFF в других случаях.

**Меры предосторожности**

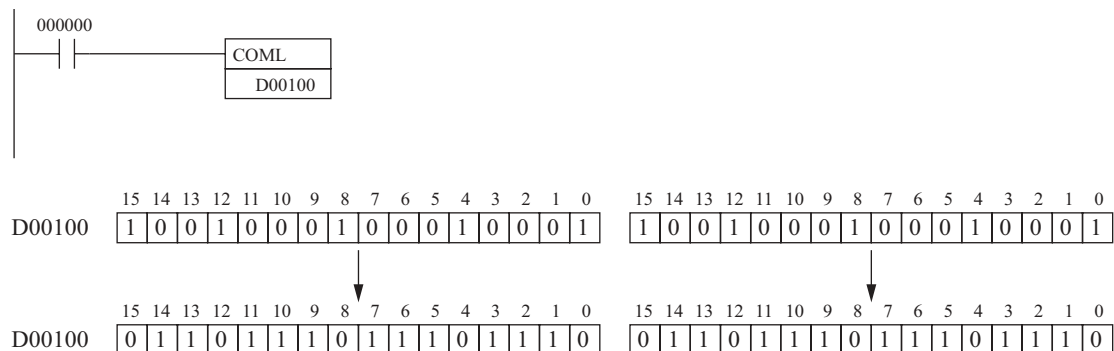
После выполнения команды COML(614) флаг ошибки переводится в состояние OFF.

Когда в результате выполнения команды COML(614) содержание результирующих слов R, R+1 равно 00000000 (шестн.) флаг равенства переводится в состояние ON.

Когда в результате выполнения команды старший бит результирующего слова R+1 равен 1, флаг отрицательного значения переводится в состояние ON.

**Примеры**

Когда в следующем ниже примере CIO 000000 находится в состоянии ON, состояние каждого из битов в словах D00100 и D00101 будет изменяться на противоположное состояние.



## 3-14 Специальные математические команды

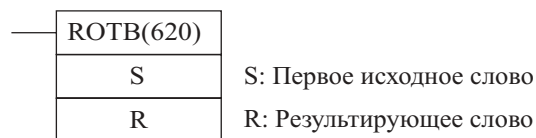
Настоящий раздел описывает команды, применяемые для выполнения специальных математических вычислений.

Команда	Мнемоническое изображение	Код функции	Страница
BINARY ROOT	ROTB	620	400
BCD SQUARE ROOT	ROOT	072	401
ARITHMETIC PROCESS	APR	069	405
FLOATING POINT DIVIDE	FDIV	079	410
BIT COUNTER	BCNT	067	414

## 3-14-1 Команда вычисления квадратного корня из двоичного числа BINARY ROOT: ROTB(620)

**Назначение**

По команде ROTB(620) производится вычисление квадратного корня из 32-х битового двоичного числа со знаком (положительного значения), содержащегося в указанных словах. Целая часть результата записывается в указанное результирующее слово.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	ROTB(620)
	Выполняется один раз для дифференцирования вверх	@ ROTB(620)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

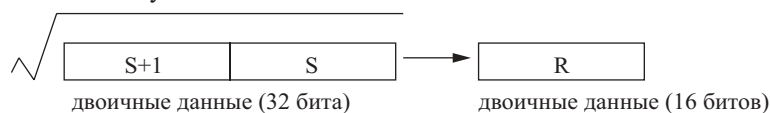
**Спецификации операндов**

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	CIO 0000...CIO 6143
Рабочая область	W000...W510	W000...W511
Область удержания бита (Holding bit area)	H000...H510	H000...H511
Область вспомогательных битов	A448...A958	A448...A959
Область таймера	T0000...T4094	T0000...T4095
Область счетчика	C0000...C4094	C0000...C4095
Область DM	D00000...D32766	D00000...D32767
Область EM, не содержащая банков	E00000...E32766	E00000...E32767
Область EM, содержащая банки	En_00000...En_32766 (n = 0... C)	En_00000...En_32767 (n = 0... C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767(n = 0... C)	
Константы	#00000000...#FFFFFFF (двоичные)	—
Регистры данных	DR0...DR15	
Индексные регистры	—	

Область	S	R
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(++) ,(- -)IR0...(- -)IR15	

**Описание**

По команде ROTB(620) производится вычисление квадратного корня из 32-х битового двоичного числа, содержащегося в словах S+1 и S. Целая часть результата записывается в результирующее слово R. Дробная часть опускается.



Диапазон значений, которые могут указываться в словах S+1 и S, - от 00000000 до 3FFFFFFF. В случае, когда указывается число от 40000000 до 7FFFFFFF, для вычисления квадратного корня используется значение 3FFFFFFF. Если заданное число превышает значение, равное 7FFFFFFF, т.е. бит 15 слова S+1 равен 1, определяется ошибка.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда бит 15 слова S+1 равен 1. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды содержание R равно нулю. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, когда содержание S+1 и S находится в пределах от 40000000 до 7FFFFFFF. OFF в других случаях.
Флаг отрицательного переполнения	UF	OFF
Флаг отрицательного значения	N	OFF

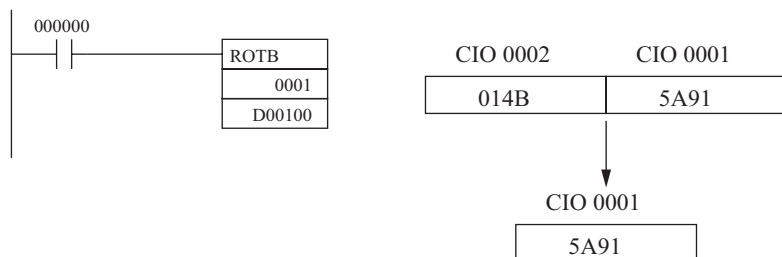
**Меры предосторожности**

Содержание S+1 и S не должно превышать значения, равного 80000000.

Операнды в данной команде (S+1, S, R) используются в виде двоичных чисел. Если исходные данные выражены в двоично-десятичном коде, используйте команду ROOT (072).

**Примеры**

Когда в следующем ниже примере CIO 000000 находится в состоянии ON, по команде ROTB(072) производится вычисление квадратного корня из данных, содержащихся в CIO 0002 и CIO 0001. Целая часть результата записывается в D00100.



### 3-14-2 Команда вычисления квадратного корня из двоично-десятичного числа BCD SQUARE ROOT: ROOT (072)

**Назначение**

По команде ROOT(072) производится вычисление квадратного корня из восьмизначного двоично-десятичного числа. Целая часть результата записывается в указанное результирующее слово.



**Символ релейно-контактной схемы**

—	ROOT(072)	
	S	S: Первое исходное слово
	R	R: Результирующее слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	ROOT (072)
	Выполняется один раз для дифференцирования вверх	@ ROOT (072)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

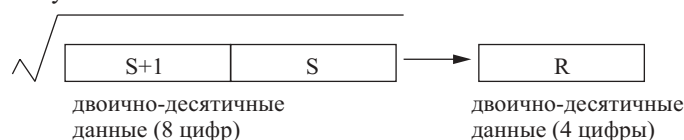
**Спецификации операндов**

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	CIO 0000...CIO 6143
Рабочая область	W000...W510	W000...W511
Область удержания бита (Holding bit area)	H000...H510	H000...H511
Область вспомогательных битов	A448...A958	A448...A959
Область таймера	T0000...T4094	T0000...T4095
Область счетчика	C0000...C4094	C0000...C4095
Область DM	D00000...D32766	D00000...D32767
Область EM, не содержащая банков	E00000...E32766	E00000...E32767
Область EM, содержащая банки	En_00000...En_32766 (n = 0... C)	En_00000...En_32767 (n = 0... C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767(n = 0... C)	
Константы	#00000000...#99999999 (двоичные)	—
Регистры данных	—	DR0...DR15
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15	

**Описание**

По команде ROOT(072) производится вычисление квадратного корня из восьмизначного двоично-десятичного числа, содержащегося в словах S+1 и S. Целая часть результата записывается в результирующее слово R. Дробная часть опускается. Дробная часть результата опускается.

Рисунок



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда данные слов S+1 и S выражены не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.

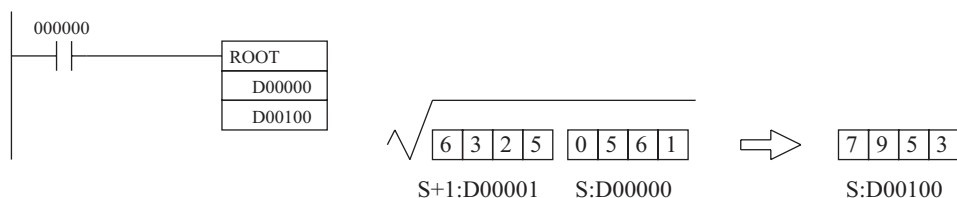
**Меры предосторожности**

Операнды в данной команде (S+1, S, R) используются в виде чисел в двоично-десятичном коде. Если исходные данные выражены в двоичном коде, используйте команду ROTB(620).

**Примеры****Вычисление квадратного корня из восьмизначного числа**

Когда в следующем ниже примере СЮ 000000 находится в состоянии ON, по команде ROOT(072) производится вычисление квадратного корня из данных, содержащихся в D00001 и D00000. Целая часть результата записывается в D00100.

**Примечание:** При вычислениях с восьмизначным числом цифры после десятичной точки отбрасываются.

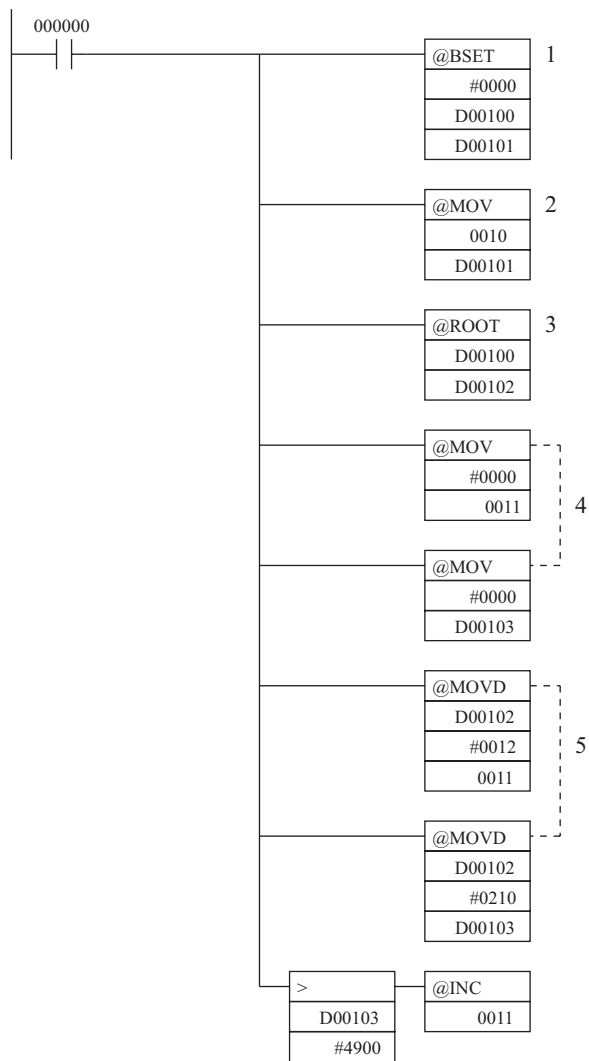
**Вычисление квадратного корня из четырехзначного числа**

Следующий ниже пример показывает порядок вычисления квадратного корня из четырехзначного числа и округления результата. Программа вычисляет квадратный корень из четырехзначного числа в СЮ 0010 и округляет результат, а также записывает результат в СЮ 0011. (Обычно, четырехзначное число умножается на 10000, и результат делится на 100, повышая точность вычислений в 100 раз.)

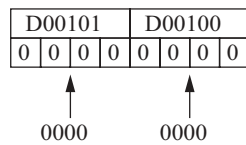
**Примечание:** При вычислениях с четырехзначным числом цифры после десятичной точки округляются.

$$\sqrt{6017} = 77,56... \rightarrow 78$$

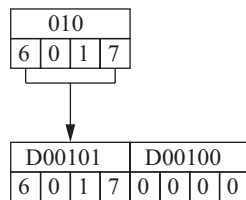
**Примечание:** Значения после десятичной точки округляются.



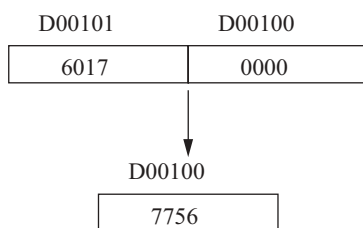
1,2,3... 1. Исходные слова (D00101 и D00100) сбрасываются в значение 0000 0000.



2. Четырехзначное число перемещается в D00101

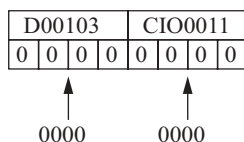


3. По команде ROOT(072) производится вычисление квадратного корня из содержания D00100 и D00100. Результат записывается в D00102.

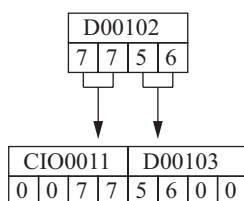


$$\sqrt{60170.000} = 7756.932\dots$$

4. В D00103 и результирующее слово СЮ 0011 записывается значение 0000 0000.

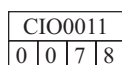


5. Результат вычисления квадратного корня делится на 100, после чего целая часть записывается в СЮ 0011, а дробная часть – в D00103.



6. Если содержание D00103 больше, чем 4900, СЮ 0011 увеличивается на 1. В этом случае результат равен 78.

$$5600 > 4900 ?$$



### 3-14-3 Команда выполнения математических операций ARITHMETIC PROCESS: APR(069)

#### Назначение

По команде APR(069) производится вычисление синуса, косинуса, или выполняется экстраполяция исходных данных. Функция линейной экстраполяции позволяет осуществлять кусочно-линейную аппроксимацию любой зависимости между X и Y.

#### Символ релейно-контактной схемы

APR(069)	
C	C: Управляющее слово
S	S: Исходное слово
R	R: Результирующее слово

#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	APR(069)
	Выполняется один раз для дифференцирования вверх	@ APR(069)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****C: Управляющее слово**

Содержание управляющего слова определяет выполняемую математическую операцию. Значением C должно быть #0000, #0001, или адрес слова.

В случае если C равно #0000, #0001, по команде APR(069) производится вычисление синуса или косинуса от S. Если C является адресом слова, по команде APR(069) производится экстраполяция значения Y от значения X, содержащегося в S, базируясь на координаты (формирующие линейные сегменты), предварительно введенные в таблицу, начинающуюся с C.

**S: Исходные данные**

Когда C равно #0000 (вычисление синуса), или #0001 (вычисление косинуса), значение S должно находиться в диапазоне от 0000 до 0900 в двоично-десятичном коде (от 0.0 до 90.0).

**Спецификации операндов**

Область	C	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
1 Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767(n = 0... C)		
Константы	Только указанные значения	–	
Регистры данных	–	DR0...DR15	
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15(++ ,(- -)IR0...,(- -)IR15		

**Описание**

Функционирование команды APR(069) зависит от содержания управляющего слова C. В случае, когда C равно 0000 или 0001, по команде APR(069) производится вычисление синуса или косинуса от S. При этом S выражено в единицах, равных одной десятой градуса.

Если C является адресом слова, по команде APR(069) производится экстраполяция значений Y от значений X, содержащихся в S, базируясь на координаты (формирующие отрезки линии), введенные предварительно в таблицу, начинающуюся с C.

**Вычисление синуса (C=0000)**

Когда C равно 0000, по команде APR(069) производится вычисление SIN(S). Результат записывается в R. Диапазон значений S – от 0000 до 0900 в двоично-десятичном коде (от 0.0 до 90.0), а диапазон значений R – от 0000 до 9999 в двоично-десятичном коде (0.0000 до 0.9999). Значения далее четвертой цифры после запятой отбрасываются.

**Вычисление синуса (C=0001)**

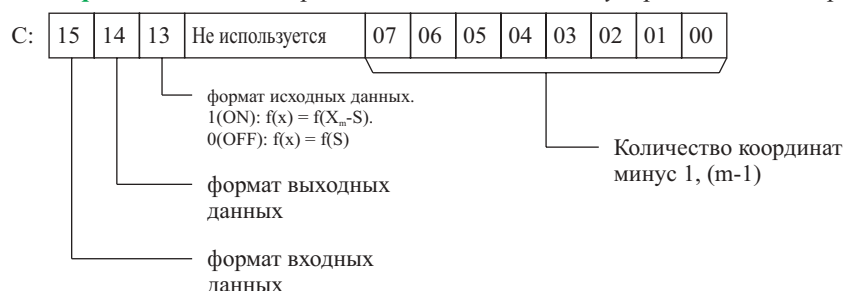
Когда C равно 0000, по команде APR(069) производится вычисление COS(S). Результат записывается в R. Диапазон значений S – от 0000 до 0900 в двоично-десятичном коде (от 0.0 до 90.0), а диапазон значений R – от 0000 до 9999 в двоично-десятичном коде (0.0000 до 0.9999). Значения далее четвертой цифры после запятой отбрасываются.

**Линейная экстраполяция**

Если содержание S является адресом слова, по команде APR(069) производится линейная экстраполяция. Содержание слова C указывает количество координат в таблице данных, начинающейся с C+2, форму исходных данных, кодировку данных (двоичные или двоично-десятичные). Биты от 00 до 07 содержат количество (двоичное) координат линий минус 1, m-1. Биты от 08 до 12 не используются. Бит 13 указывает либо зависимость  $f(x) = f(S)$ , либо зависимость  $f(x) = f(X_m - S)$ , при этом состоянию OFF соответствует  $f(x)$

$= f(S)$ , а состоянию ON -  $f(x) = f(X_m - S)$ . Бит 14 определяет кодировку выходных данных, а бит 15 – кодировку входных данных (двоичные или двоично-десятичные данные). Состояние OFF задает двоичные данные, ON задает двоично-десятичные данные.

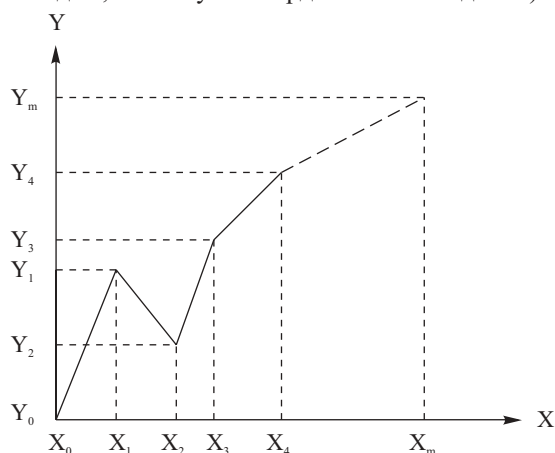
**Примечание:** Координаты  $X$  должны быть упорядочены по возрастанию:  $X_1 < X_2 < \dots < X_m$ .



В следующей ниже таблице показаны функции битов слова С.

Биты слова С	Функция
07...00	Вводите $m-1$ , т.е. количество линейных отрезков ( $m$ ) минус 1.
08...12	Не используются. Биты устанавливаются в значение 0.
13	Обработка входных битов: 0: S используется в качестве входных данных, в том виде, как оно есть. 1: В качестве входных данных используется $X_m - S$ . $X_m$ – это максимальное значение $X$ .
14	Формат входных данных (0: двоичные данные.1: двоично-десятичные данные).
15	Формат входных данных (0: двоичные данные.1: двоично-десятичные данные).

В следующей ниже таблице показана функция битов в словах  $C+1 \dots C+2m=2$ . Введите координаты конечной точки  $m+1$ , которая определяет  $m$  линейных отрезков. Все координаты вводите в двоичном коде. ( $X_0$  всегда 0, поэтому ее координаты не вводятся.)



Слово	Координата
C+1	$X_m$ (Максимальное значение $X$ )
C+2	$Y_0$
C+3	$X_1$
C+4	$Y_1$
C+5	$X_2$
C+6	$Y_2$
...	...
C+(2m+1)	$X_m$ (Максимальное значение $X$ )
C+(2m+2)	$Y_m$

- $Y_n = f(X_n)$ ,  $Y_0 = f(X_0)$
- Убедитесь в том, что во всех случаях справедливо неравенство  $X_{n-1} < X_n$ .
- Вводите все значения ( $X_n$ ,  $Y_n$ ) в виде двоичных данных.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда С является константой, больше чем 0001. Переводится в состояние ON, когда С является адресом слова, однако координаты X не упорядочены по возрастанию (не соответствуют неравенству $X_1 < X_2 < \dots < X_m$ ). Переводится в состояние ON, когда С является адресом слова, бит 15 слова С указывает двоично-десятичные входные данные, однако содержание слова S выражено не в двоично-десятичном коде. Переводится в состояние ON, когда содержанием С является 0000 или 0001, однако содержание S не является двоично-десятичным числом в пределах от 0000 до 0900. e. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда бит 15 слова R находится в состоянии ON. OFF в других случаях.

**Меры предосторожности**

Несмотря на то, что в действительности результатом вычисления SIN(90) и COS(90) является 1, в результирующее слово R записывается значение 9999 (0.9999).

В случае, когда С является константой, больше чем 0001, определяется ошибка.

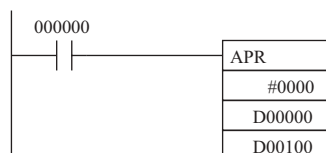
Когда при задании линейной экстраполяции координаты X не упорядочены по возрастанию (не соответствуют неравенству  $X_1 < X_2 < \dots < X_m$ ), определяется ошибка.

Когда при задании линейной экстраполяции указываются входные данные в двоично-десятичном коде (бит 15 слова С равен 1), а содержание S выражено не в двоично-десятичном коде, определяется ошибка.

Когда при задании выполнения тригонометрической функции (С=0000 или 0001) содержание исходного слова не является двоично-десятичным числом в пределах от 0000 до 0900, определяется ошибка.

**Пример****Вычисление синуса (С: #0000)**

В следующем примере показано использование команды APR(069) для вычисления синуса 30°.

**Исходные данные**

S: D00000			
0	$10^1$	$10^0$	$10^{-1}$
0	3	0	0

Указывайте исходные данные в единицах, равных  $10^{-1}$  градусов (от 0000 до 0900 в двоично-десятичном коде).

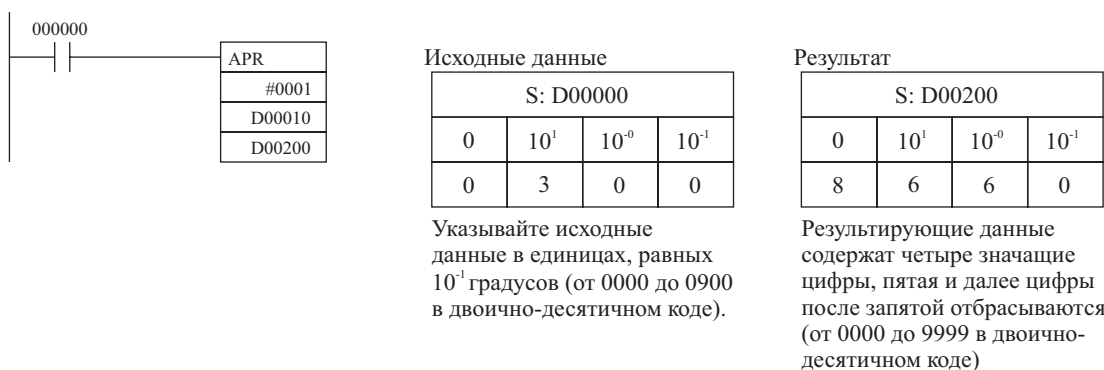
**Результат**

S: D00100			
0	$10^1$	$10^0$	$10^{-1}$
5	0	0	0

Результирующие данные содержат четыре значащие цифры, пятая и далее цифры после запятой отбрасываются (от 0000 до 9999 в двоично-десятичном коде)

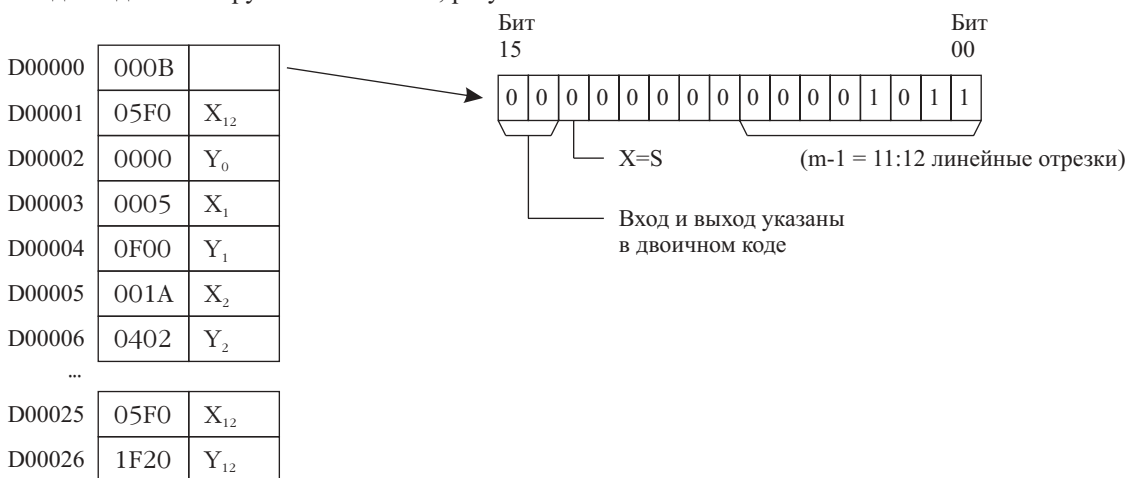
**Вычисление косинуса (С: #0001)**

В следующем примере показано использование команды APR(069) для вычисления косинуса 30°. (COS(30)=0.8660)

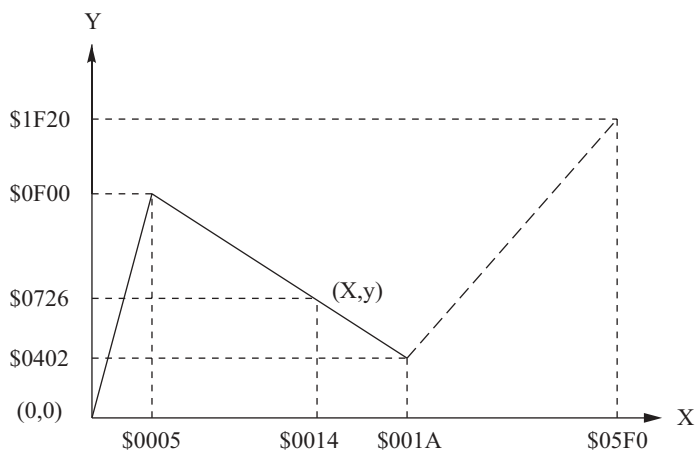


**Линейная экстраполяция (С: адрес слова)**

В следующем примере показано построение экстраполирующей ломаной линии, имеющей 12 координат. Блок данных достаточно продолжителен, как это и должно быть, от D00000 до D00026 (C...C+(2...12+2)). Входные данные берутся из СЮ 0010, результат записывается в СЮ 0011.



В данном случае исходное слово СЮ 0010 содержит значение 0014, и f(0014)= 0728 записывается в R, СЮ 0011.



Ниже приводится порядок вычислений для линейной экстраполяции.

$$Y = 0F00 + \frac{0402 - 0F00}{001A - 0005} \times (0014 - 0005) = 0F00 - (0086 \times 000F) = 0726$$

Все значения указаны в шестнадцатеричном коде.



## 3-14-4 Команда выполнения деления чисел с плавающей запятой FLOATING POINT DIVIDE: FDIV(079)

**Назначение**

По команде FDIV(079) производится деление двух семизначных чисел с плавающей запятой. Числа с плавающей запятой выражены в экспоненциальном виде (7 знаков мантисса и один знак показатель степени)

**Символ релейно-контактной схемы**

FDIV(079)	
Dd	Dd: Первое слово делимого
Dr	Dr: Первое слово делителя
R	R: Первое слово результата

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	FDIV(079)
	Выполняется один раз для дифференцирования вверх	@ FDIV(079)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

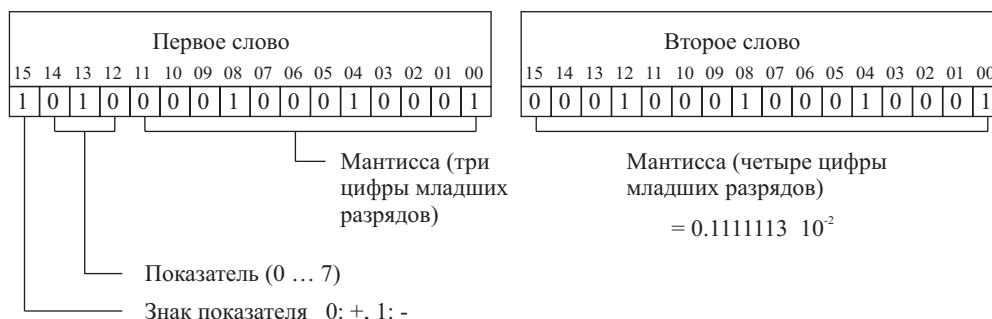
Область	Dd	Dr	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142		
Рабочая область	W000...W510		
Область удержания бита (Holding bit area)	H000...H510		
Область вспомогательных битов	A000...A958	A448...A958	
Область таймера	T0000...T4094		
Область счетчика	C0000...C4094		
Область DM	D00000...D32766		
Область EM, не содержащая банков	E00000...E32766		
Область EM, содержащая банки	En_00000...En_32766 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767(n = 0... C)		
Константы	-		
Регистры данных	-		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15-2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15		

**Описание**

По команде FDIV(079) выполняется деление чисел с плавающей запятой, находящихся в Dd и Dd+1 на числа с плавающей запятой, находящиеся в Dr и Dr+1. Результат (частное) записывается в R и R+1.

Dd+1	Dd	Dr+1	Dr
Частное			
R+1		R	

Для представления чисел с плавающей запятой 7 младших разрядов используются для записи мантиссы, а старший разряд - для записи показателя степени, как показано на следующем ниже рисунке. Диапазон изменения старшей цифры - от 0 до F. При этом положительные значения показателя - от 0 до 7, а отрицательные значения - от 0 до F (0...-7). Цифры в семи младших разрядах должны быть выражены в двоично-десятичном коде.



Еще два примера чисел с плавающей запятой.

61234567: 0.1234567...10<sup>6</sup> (6 = 0110 в двоичном коде).

B1234567: 0.1234567...10<sup>-3</sup> (B = 1011 в двоичном коде).

В следующей ниже таблице представлены максимально допустимые и минимально допустимые значения.

Предел	Восьмизначные шестнадцатеричные	Числа с плавающей запятой
Максимальное значение	79999999	0.9999999...10 <sup>7</sup>
Минимальное значение (делимое и делитель)	F0000001	0.0000001...10 <sup>-7</sup>
Минимальное значение (частное)	F1000000	0.1000000...10 <sup>-7</sup>

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда мантисса в словах Dd+1 и Dd (7 цифр младших разрядов) выражена не в двоично-десятичном коде. Переводится в состояние ON, когда мантисса в словах Dr+1 и Dr (7 цифр младших разрядов) выражена не в двоично-десятичном коде. Переводится в состояние ON, когда делителем (Dr+1 и Dr) является 0. Переводится в состояние ON, когда содержание результирующих слов выходит за пределы от 0.0000001...10 <sup>-7</sup> до 0.9999999...10 <sup>7</sup> . OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен нулю. OFF в других случаях.

#### Меры предосторожности

Результат выражен числом с плавающей запятой, поэтому он содержит 7 значащих цифр. Восьмая и далее цифры отбрасываются.

Диапазон значений результата должен находиться в пределах от 0.0000001...10<sup>-7</sup> до 0.9999999...10<sup>7</sup>.

#### Пример

##### Базовое деление с плавающей запятой

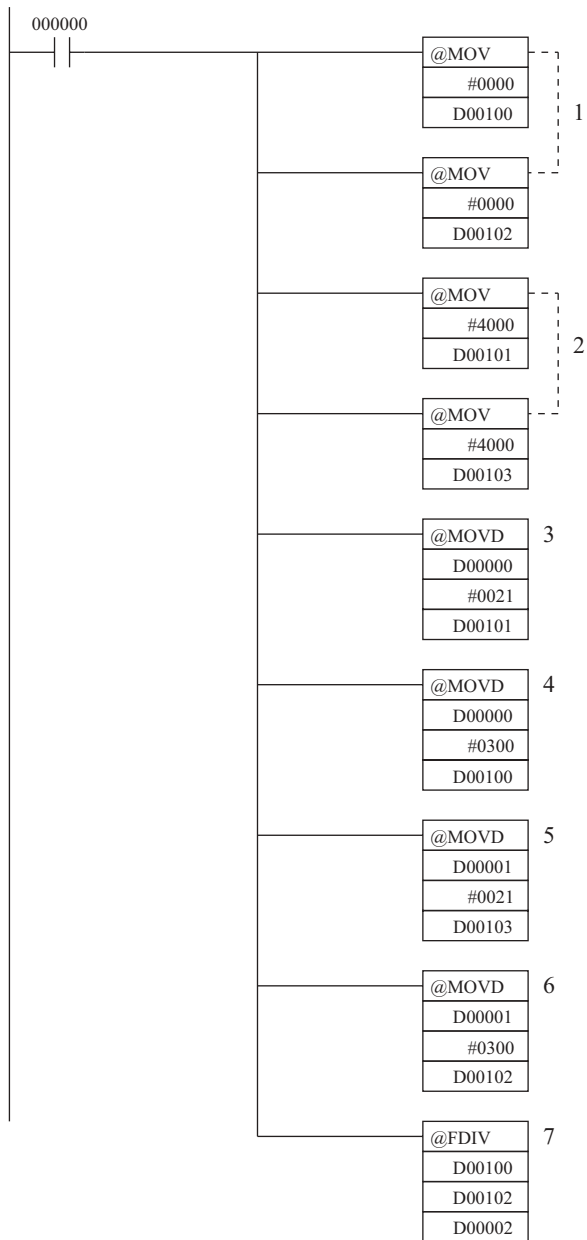
В следующем примере, когда CIO 000000 находится в состоянии ON, по команде FDIV(079) производится деление числа с плавающей запятой, находящегося в D00101 и D00100, на числа с плавающей запятой, находящиеся в CIO 0021 и CIO 0020. Результат записывается в D00301 и D00300.

FDIV	D00100	D00101	D00100	0.5670000x10 <sup>-2</sup>
	0020	A567	0000	
	D0300	CIO0021	CIO0020	0.1234567x10 <sup>-3</sup>
		B123	4567	
		D00301	D00300	0.4592703x10 <sup>2</sup>
		2459	2703	

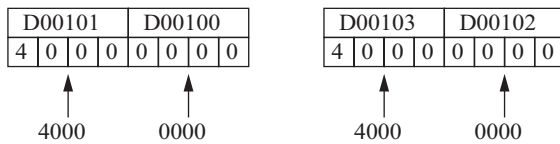
**Деление с плавающей запятой двух двоично-десятичных чисел**

В данном примере четырехзначное двоично-десятичное число в D00000 делится на четырехзначное двоично-десятичное число в D00001. Результат в виде числа с плавающей запятой записывается в D00003 и D00002.

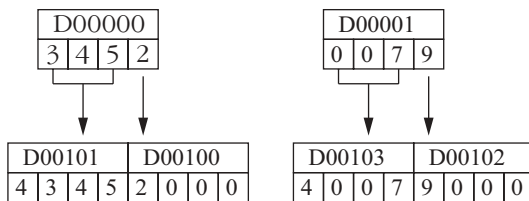
Для выполнения деления чисел с плавающей запятой двоично-десятичное число, находящееся в D00000, преобразуется в формат числа с плавающей запятой в D00101 и D00100, а двоично-десятичное число, находящееся в D00001, преобразуется в формат числа с плавающей запятой в D00103 и D00102.



- 1,2,3... 1. D00100 и D00102 устанавливаются в значение, равное 0000.  
 2. D00101 и D00103 устанавливаются в значение, равное 4000.



3. Для перемещения цифр исходного слова в формат двойного слова числа с плавающей запятой используется команда MOVD(083).



4. По команде *FDIV(079)* производится деление числа с плавающей запятой, находящегося в *D00101* и *D00100*, на число с плавающей запятой, находящееся в *D00103* и *D00102*.

### 3-14-5 Команда подсчета битов BIT COUNTER: BCNT(067)

#### Назначение

По команде BCNT(067) производится подсчет общего количества битов, находящихся в состоянии ON в указанном слове (словах).

#### Символ релейно-контактной схемы

—	BCNT(067)	
	N	N: Номера слов
	S	S: Персвое исходное слово
	R	R: Результирующее слово

#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	BCNT(067)
	Выполняется один раз для дифференцирования вверх	@ BCNT(067)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

##### N: Номера слов

Номера указываемых слов должны находиться в пределах от 0001 до FFFF (от 1 до 65535 слов).

##### S: Первое исходное слово

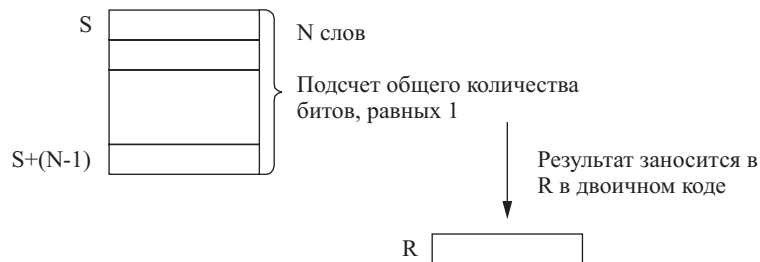
Слова S и S+1 должны находиться в одной области данных.

#### Спецификации операндов

Область	N	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
1 Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767(n = 0... C)		
Константы	#0001...#FFFF (двоичные) или &1...&65535	—	
Регистры данных	DR0...DR15	—	DR0...DR15
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15		

**Описание**

По команде BCNT(067) производится подсчет общего количества битов, находящихся в состоянии ON, во всех словах между S и S+(N-1). Результат записывается в R.

**Флаги**

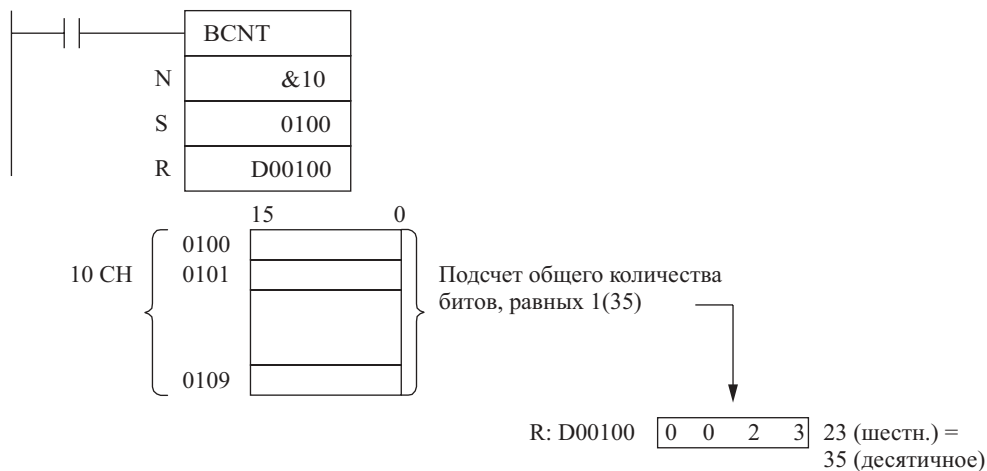
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда N = 0000. Переводится в состояние ON, когда результат превышает значение, равное FFFF. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен 0000. OFF в других случаях.

**Меры предосторожности**

Когда N = 0000 или результат превышает значение, равное FFFF, определяется ошибка.

**Пример**

В следующем примере, когда CIO 000000 находится в состоянии ON, по команде BCNT(067) производится подсчет общего количества битов, находящихся в состоянии ON, в десяти словах от CIO 0100 до CIO 0109. Результат записывается в D00100.



## 3-15 Команды математических операций с плавающей запятой

Перечисленные ниже команды осуществляют преобразование данных и производят математические операции с плавающей запятой. Модули центрального процессора серии CS1 поддерживают приведенные в таблице команды.

Команда	Мнемоническое изображение	Код функции	Страница
FLOATING TO 16-BIT	FIX	450	421
FLOATING TO 32-BIT	FIXL	451	422
16-BIT TO FLOATING	FLT	452	424
32-BIT TO FLOATING	FLTL	453	425
FLOATING-POINT ADD	+F	454	427
FLOATING-POINT SUBTRACT	-F	455	428
FLOATING-POINT MULTIPLY	*F	456	430
FLOATING-POINT DIVIDE	/F	457	432
DEGREES TO RADIANS	RAD	458	434
RADIANS TO DEGREES	DEG	459	436
SINE	SIN	460	437
COSINE	COS	461	439
TANGENT	TAN	462	440
ARC SINE.	ASIN	463	442
ARC COSINE	ACOS	464	444
ARC TANGENT	ATAN	465	446
SQUARE ROOT	SQRT	466	447
EXPONENT	EXP	467	449
LOGARITHM	LOG	468	451
EXPONENTIAL POW	PWR	840	453

**Формат данных**

Данные с плавающей запятой выражают действительные числа, используя знак, показатель степени и мантиссу. При использовании формы выражения числа с плавающей запятой применяется следующая формула.

Действительное число =  $(-1)^s 2^{e-127} (1.f)$

$Rn = (-1)^n 2^{e-127} (1.f)$

Где: Rn - Действительное число

s - знак

e - показатель

f - мантисса.

Формат данных с плавающей запятой подчиняется стандартам IEEE754. Данные состоят из 32-х битов, как показано ниже.

Знак      Показатель                      Мантисса

S	e	f
31	30	23 22
		0

Данные	Количество битов	Содержание
s: знак	1	0: положительный. 1: отрицательный.
e: показатель	8	Диапазон значений показателя степени (e) - от 0 до 255. Действительное значение показателя равно значению, оставшемуся после вычитания 127 из e. Таким образом, реальное значение показателя от -127 до 128. Значения «e =0» и «e =255» выражают специальные числа.
f: мантисса	23	Мантисса двоичных данных с плавающей запятой соответствует соотношению $2.0 > 1.f > 1.0$ .

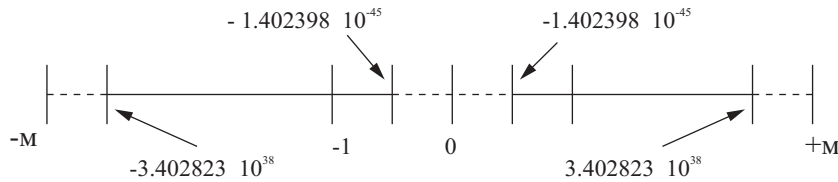
**Количество знаков**

Эффективным количеством знаков для данных с плавающей запятой является 24 бита (примерно семь десятичных цифр).

**Данные с плавающей запятой**

При помощи данных с плавающей запятой можно выразить следующие данные:

- $-\infty$
- $-3.402823 \cdot 10^{38} \leq \text{значение} \leq -1.402398 \cdot 10^{-45}$
- 0
- $-1.402398 \cdot 10^{-45} \leq \text{значение} \leq 3.402823 \cdot 10^{38}$
- $+\infty$
- Неопределенность (NaN)



### Специальные числа

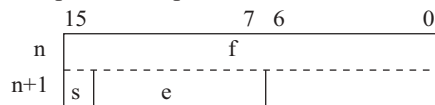
Форматами для NaN,  $+\infty$  и  $-\infty$  являются:

- NaN:  $e = 255, f = 0$
- $+\infty$ :  $e = 255, f = 0, s = 0$
- $-\infty$ :  $e = 255, f = 0, s = 1$
- 0:  $e = 0$

**Примечание:** NaN (неопределенность) не является в действительности числом с плавающей запятой. Выполнение операций с плавающей запятой никогда не приведет к результату NaN.

### Запись данных с плавающей запятой

Когда для дисплея редактирования памяти ввода/вывода в СХ-программаторе указываются данные в форме числа с плавающей запятой, обычные десятичные данные, вводимые в дисплей, автоматически преобразуются в формат чисел с плавающей запятой (согласно стандарту IEEE754) и записываются в память ввода/вывода. Данные, записанные в формате IEEE754, автоматически преобразуются в десятичные числа при мониторинге с помощью дисплея.



При чтении и записи данных с плавающей запятой пользователю нет необходимости беспокоиться о преобразовании данных в формат IEEE754. Необходимо только помнить, что числа с плавающей запятой занимают два слова.

### Числа, выражаемые в виде значений с плавающей запятой

В работе могут использоваться следующие числа с плавающей запятой.

Мантисса (f)	Показатель степени		
	0	Не 0 и не все единицы	Все единицы (255)
0	0	Обычное число	Бесконечность
Не 0	Необычное число		NaN

**Примечание:** Необычное число – это число, абсолютное значение которого так мало, что его невозможно выразить как обычное число. Такое число имеет очень мало значащих цифр. Если результатом вычисления является необычное число (включая промежуточные результаты), количество значащих цифр уменьшается.

### Обычные числа

Обычные числа выражают действительные числа. Для положительных значений бит знака равен нулю, а для отрицательных чисел бит знака равен единице.

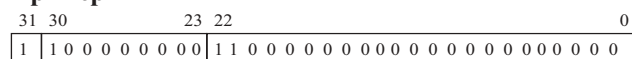
Показатель (e) имеет значение от 1 до 254, однако действительное значение показателя будет на 127 меньше, т.е. от -126 до 127.

Мантисса (f) имеет значения от 0 до  $2^{33} - 1$ , и предполагается, что в реальной мантиссе бит  $2^{33}$  равен 1, а десятичная точка расположена непосредственно после него.

Обычные числа выражаются следующим образом:

$$(-1)^s 2^{e-127} \times (1 + \text{мантисса} \times 2^{-23})$$

#### Пример





Знак:	- (минус)
Показатель степени:	$128 - 127 = 1$
Мантисса:	$1 + (2^{22} + 2^{21}) \times 2^{-23} = 1 + 2^{-1} + 2^{-2} = 1 + 0.75 = 1.75$
Значение:	$-1.75 \times 2^1 = -3.5$

**Необычные числа**

Необычные числа выражают действительные числа с очень маленькими абсолютными значениями. Для положительного значения бит знака равен 0, для отрицательного значения бит знака равен 1.

Когда показатель степени (e) равен 0, его действительное значение будет равно  $-126$ .

Мантисса (f) имеет значения от 0 до  $2^{33} - 1$ , и предполагается, что в реальной мантиссе бит  $2^{33}$  равен 0, а десятичная точка расположена непосредственно после него.

Необычные числа выражаются следующим образом:

$$(-1)^{(s)} \times 2^{-126} \times (\text{мантисса} \times 2^{-23})$$

**Пример**

31	30	23	22	0
0	0	0	0	0
0	0	0	0	0
1	1	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Знак:	- (минус)
Показатель степени:	$-126$
Мантисса:	$0 + (2^{22} + 2^{21}) \times 2^{-23} = 0 + (2^{-1} + 2^{-2}) = 0 + 0.75 = 0.75$
Значение:	$-0.75 \times 2^{-126}$

**Ноль**

Значения, равные  $+0.0$  и  $-0.0$  могут выражаться посредством установки бита знака в состояние 0 для положительных значений, или в состояние 1 для отрицательных значений. Показатель степени и мантисса в этом случае равны нулю. Оба значения,  $+0.0$  и  $-0.0$ , эквивалентны значению 0.0. Для ознакомления с различием, обусловленным установкой знака перед значением 0.0 обратитесь ниже, к подразделу «**Результаты выполнения операций с плавающей запятой**».

**Бесконечность**

Значения  $+\infty$  и  $-\infty$  могут выражаться посредством установки бита знака в состояние 0 для положительных значений, или в состояние 1 для отрицательных значений. Показатель степени в этом случае будет равен 255 ( $2^8 - 1$ ) а мантисса равна нулю.

**Неопределенность (Не число)**

NaN (неопределенность) определяется в том случае, когда результат вычислений, например таких как  $0.0/0.0$ , / , - , не соответствует числу или бесконечности. В этом случае показатель будет равен 255 ( $2^8 - 1$ ), а мантисса не равна нулю.

*Примечание:* Не существует определения для знака перед NaN, или для диапазона значений мантиссы (кроме того, что мантисса не равна 0).

**Результаты выполнения операций с плавающей запятой****Округление результатов**

Когда количество цифр результата после выполнения операций с плавающей запятой превышает количество значащих цифр для внутренних вычислений, для округления чисел используются следующие методы.

Если результат близок к одному или двум внутренним выражениям с плавающей запятой, используется ближайшее выражение. Если результат одинаково близок к двум внутренним выражениям, результат округляется таким образом, что последняя цифра мантиссы становится равной нулю.

**Переполнения, отрицательные переполнения и недействительные вычисления**

Сигнал переполнения подается в виде положительной или отрицательной бесконечности, в зависимости от знака результата. Сигнал отрицательного переполнения подается в виде положительного или отрицательного нуля, в зависимости от знака результата.

Результатом недействительных вычислений является NaN (неопределенность). К недействительным операциям относятся действия, подобные сложению бесконечности с числом, имеющим противоположный знак, вычитанию бесконечности из числа с противоположным знаком, умножению нуля и бесконечности, делению нуля на ноль, делению бесконечности на бесконечность.

Результат может быть неправильным, если в процессе преобразования числа с плавающей запятой в целое число возникает переполнение.

**Меры предосторожности при обращении с особыми величинами**

При обращении с такими величинами, как ноль, бесконечность или неопределенность предпринимайте следующие меры предосторожности.

Суммой «положительного» нуля и «отрицательного» нуля является «положительный» ноль.

Разницей между нулями, имеющими одинаковые знаки, является «положительный» ноль.

Если одним из операндов является NaN (неопределенность), результатом является NaN (неопределенность).

Ноль с отрицательным знаком и ноль с положительным знаком при сравнении считаются эквивалентными.

Сравнительный контроль или контроль эквивалентности одного или более выражений NaN (неопределенность) всегда являются истинными для !=, и всегда являются ложными для любых других команд.

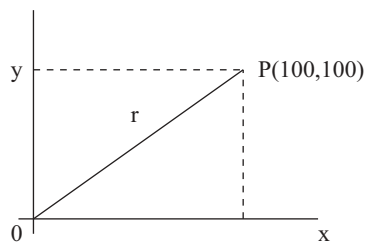
#### Результаты вычислений с плавающей запятой

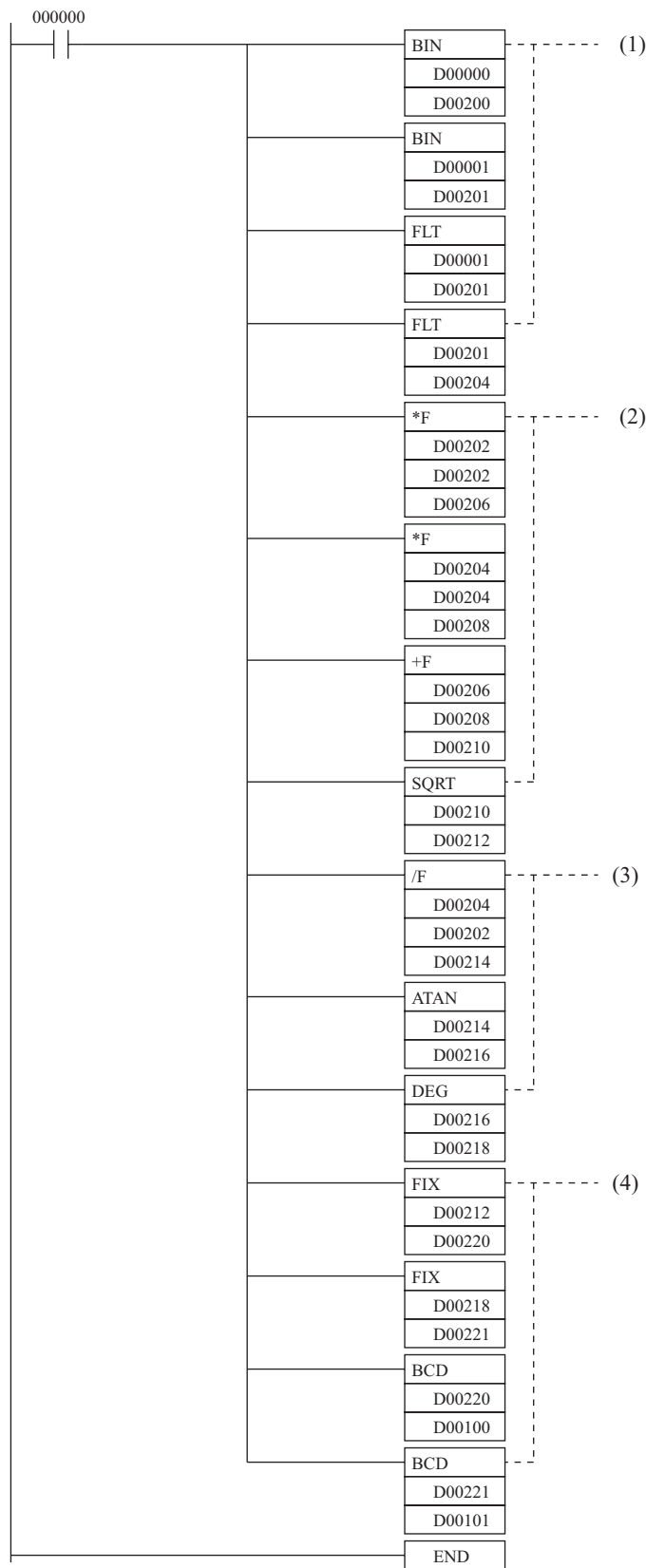
Когда абсолютное значение результата больше, чем максимально возможное для выражения в виде данных с плавающей запятой, флаг переполнения переводится в состояние ON, а результат выводится в виде . Если результат положителен на выход выводится + , если результат отрицателен, выводится - .

Когда в результате вычисления показатель (e) и мантисса (f) равны нулю, только флаг равенства переводится в состояние ON. Результат вычислений выводится как нулевое значение, когда абсолютное значение результата меньше минимально возможного для выражения в виде числа с плавающей запятой. В этом случае флаг отрицательного переполнения переводится в состояние ON.

#### Пример

В данном примере программы координаты X и Y задаются четырехзначными двоично-десятичными числами в D00000 и D00001. Расстояние (r) от начала координат и угол ( в градусах) определяются и записываются в D00100 и D00101. Цифры, расположенные после десятичной точки, отбрасываются. (Результат округляется до целого числа.)





**Вычисления**

Расстояние  $r = \sqrt{x^2 + y^2}$ , например  $r = \sqrt{100^2 + 100^2} = 141.4214$

Угол  $\theta = \tan^{-1}\left(\frac{y}{x}\right)$ , например  $\theta = \tan^{-1}\left(\frac{100}{100}\right) = 45.0$

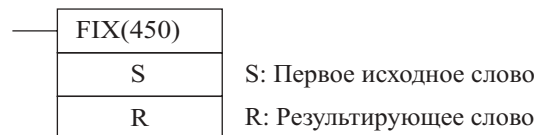
1. Данный раздел программы осуществляет преобразование двоично-десятичных данных в данные с плавающей запятой.
  - a. Область данных, начиная с D00200 и далее, используется в качестве рабочей области.
  - b. Первая команда BIN(023) используется для временного преобразования двоично-десятичных данных в двоичные данные, затем команда FLT(452) используется для преобразования двоичных данных в данные с плавающей запятой.
  - c. Значение x, преобразованное в данные с плавающей запятой, выводится в D00203 и D00202.
  - d. Значение Y, преобразованное в данные с плавающей запятой, выводится в D00205 и D00204.
2. Для вычисления расстояния применяются команды выполнения операций с плавающей запятой, осуществляющие вычисление квадратного корня из выражения  $(x^2 + y^2)$ . Результат записывается в D00213 и D00212 в виде данных с плавающей запятой.
3. Для вычисления угла применяются команды выполнения операций с плавающей запятой, осуществляющие вычисление функции  $\tan^{-1}(x/y)$ . По команде ATAN(465) результат определяется в радианах, поэтому для преобразования результата в градусы применяется команда DEG(459). После этого результат записывается в D00219 и D00218 в виде данных с плавающей запятой.
4. Производится преобразование данных из чисел с плавающей запятой в двоично-десятичные значения.
  - a. Первая команда FIX(450) используется для временного преобразования данных с плавающей запятой в двоичные данные, для дальнейшего преобразования данных в двоично-десятичные данные применяется команда BCD(024).
  - b. Расстояние r записывается в D00100.
  - c. Угол записывается в D00101.

### 3-15-1 Команда преобразования данных с плавающей запятой в двоичные данные FLOATING TO 16-BIT: FIX(450)

#### Назначение

По команде FIX(450) осуществляется преобразование числа длиной 32 бита с плавающей запятой в двоичные данные длиной 16 битов со знаком. Результат записывается в указанное слово.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	FIX(450)
	Выполняется один раз для дифференцирования вверх	@ FIX(450)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

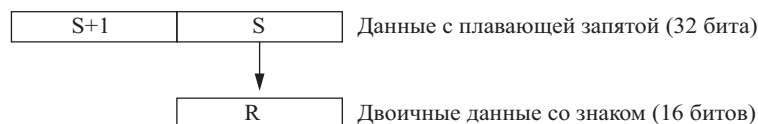
#### Спецификации операндов

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	CIO 0000...CIO 6143
Рабочая область	W000...W510	W000...W511
Область удержания бита (Holding bit area)	H000...H510	H000...H511
Область вспомогательных битов	A000...A958	A448...A959
Область таймера	T0000...T4094	T0000...T4095
Область счетчика	C0000...C4094	C0000...C4095
Область DM	D00000...D32766	D00000...D32767
Область EM, не содержащая банков	E00000...E32766	E00000...E32767

Область	S	R
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFFF (двоичные)	–
Регистры данных	–	DR0...DR15
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(-)IR0...,(-)IR15	

**Описание**

По команде FIX(450) осуществляется преобразование целой части числа длиной 32 бита с плавающей запятой, содержащегося в S+1 и S (формат IEEE754), в двоичные данные длиной 16 битов со знаком. Результат записывается в R.



Преобразованию подвергается только целая часть данных с плавающей запятой, дробная часть числа сокращается. Целая часть преобразуемого числа не должна выходить за пределы от –32768 до 32767.

**Примеры преобразования:**

Число 3.5 с плавающей запятой преобразуется в число 3.

Число -3.5 с плавающей запятой преобразуется в число -3.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда S+1 и S содержат неопределенность (NaN). Переводится в состояние ON, когда целая часть результата в S+1 и S выходит за пределы от –32768 до 32767. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен 0000. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда бит 15 результирующего слова равен 1. OFF в других случаях.

**Меры предосторожности**

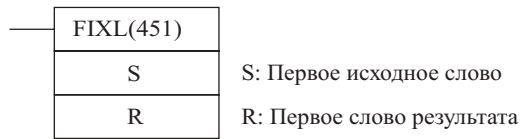
Исходные слова S+1 и S должны содержать данные с плавающей запятой, а целая часть этих данных не должна выходить за пределы от –32768 до 32767.

### 3-15-2 Команда преобразования данных с плавающей запятой в двойные двоичные данные FLOATING TO 32-BIT: FIXL(451)

**Назначение**

По команде FIXL(451) осуществляется преобразование числа длиной 32 бита с плавающей запятой в двоичные данные длиной 32 бита со знаком. Результат записывается в указанные слова.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	FIXL(451)
	Выполняется один раз для дифференцирования вверх	@ FIXL(451)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

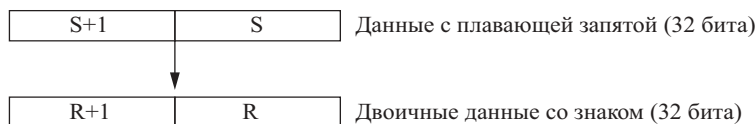
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A959
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFF (двоичные)	—
Регистры данных	—	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15	

**Описание**

По команде FIX(450) осуществляется преобразование числа длиной 32 бита с плавающей запятой, содержащегося в S+1 и S (формат IEEE754), в двоичные данные длиной 32 бита со знаком. Результат записывается в R+1 и R.



Преобразованию подвергается только целая часть данных с плавающей запятой, дробная часть числа сокращается. Целая часть преобразуемого числа не должна выходить за пределы от - 2147483648 до 2147483647.

**Примеры преобразования:**

Число 2147483640.5 с плавающей запятой преобразуется в число 2147483640.

Число - 214748340.5 с плавающей запятой преобразуется в число 214748340.

**Флаги**

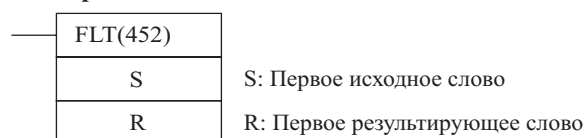
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда S+1 и S содержат неопределенность (NaN). Переводится в состояние ON, когда целая часть результата в S+1 и S выходит за пределы от – 2147483648 до 2147483647. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения команды результат равен 00000000. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда бит 15 результирующего слова R+1 равен 1. OFF в других случаях.

**Меры предосторожности**

Исходные слова S+1 и S должны содержать данные с плавающей запятой, а целая часть этих данных не должна выходить за пределы от – 2147483648 до 2147483647.

**3-15-3 Команда преобразования двоичных данных в данные с плавающей запятой 16-BIT TO FLOATING: FLT(452)****Назначение**

По команде FLT(452) осуществляется преобразование двоичного числа длиной 16 битов со знаком в данные длиной 32 бита с плавающей запятой. Результат записывается в указанные слова.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	FLT(452)
	Выполняется один раз для дифференцирования вверх	@ FLT(452)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6142
Рабочая область	W000...W511	W000...W510
Область удержания бита (Holding bit area)	H000...H511	H000...H510
Область вспомогательных битов	A000...A959	A448...A958
Область таймера	T0000...T4095	T0000...T4094
Область счетчика	C0000...C4095	C0000...C4094
Область DM	D00000...D32767	D00000...D32766
Область EM, не содержащая банков	E00000...E32767	E00000...E32766
Область EM, содержащая банки	En_00000...En_32767 (n=0...C)	En_00000...En_32766 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000 @D32767 @E00000 @E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	

Область	S	R
Константы	#0000...#FFFF (двоичные)	–
Регистры данных	DR0...DR15	–
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-(-)IR15	

**Описание**

По команде FLT(452) осуществляется преобразование двоичного числа длиной 16 битов со знаком, содержащегося в S, в двоичные данные с плавающей запятой длиной 32 бита (формат IEEE754). Результат записывается в R+1 и R.



В слове S может указываться значение, не выходящее за пределы от  $-32768$  до  $32767$ . Для преобразования данных, выходящих за указанные пределы, используйте команду FLTL(453).

**Примеры преобразования:**

Двоичное число 3 со знаком преобразуется в число 3.0.

Двоичное число -3 со знаком преобразуется в число -3.0.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда результат отрицателен. OFF в других случаях.

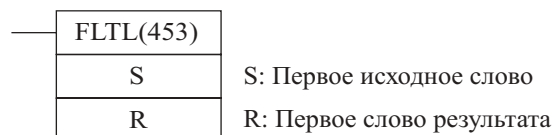
**Меры предосторожности**

Исходное слово S должно содержать двоичные данные со знаком плавающей запятой, а десятичное значение этих данных не должно выходить за пределы от  $-32768$  до  $32767$ .

### 3-15-4 Команда преобразования двоичных данных в данные с плавающей запятой 32-BIT TO FLOATING: FLTL(453)

**Назначение**

По команде FLTL(453) осуществляется преобразование двоичного числа длиной 32 бита со знаком в данные с плавающей запятой длиной 32 бита. Результат записывается в указанные слова.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	FLTL(453)
	Выполняется один раз для дифференцирования вверх	@ FLTL(453)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да



### Спецификации операндов

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFF (двоичные)	–
Регистры данных	–	
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15	

### Описание

По команде FLTL(453) осуществляется преобразование двоичного числа длиной 32 бита со знаком, содержащегося в S+1, в данные с плавающей запятой длиной 32 бита (формат IEEE754). Результат записывается слова R+1 и R. К результирующему слову с плавающей запятой добавляется один ноль после запятой.



В словах S+1 и S могут задаваться двоичные данные, не выходящие за пределы от – 2147483648 до 2147483647. Длина числа с плавающей запятой составляет 24 значащих бита. Результат может быть некорректным, если с помощью команды FLTL(453) производится преобразование числа большего, чем 16777215 (максимальное число, которое можно выразить числом, состоящим из 24-х битов).

### Примеры преобразования:

Двоичное число со знаком 16777215 преобразуется в число 16777215.0.

Двоичное число со знаком -16777215 преобразуется в число -16777215.0.\*\*\* (Опечатка Н.П.)Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, когда результат отрицателен. OFF в других случаях.

### Меры предосторожности

Результат может быть некорректным, если с помощью команды FLTL(453) производится преобразование числа с абсолютным значением, превышающим 16777215 (максимальное число, которое можно выразить числом, состоящим из 24-х битов).

3-15-5 Команда выполнения сложения двух чисел с плавающей запятой FLOATING-POINT ADD: +F(454)

**Назначение**

По команде +F(454) осуществляется сложения двух чисел длиной 32 бита с плавающей запятой. Результат записывается в указанные слова.

**Символ релейно-контактной схемы**

+F(454)	
Au	Au: Первое слово первого слагаемого
Ad	Ad: Первое слово второго слагаемого
R	R: Первое слово результата

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	+F(454)
	Выполняется один раз для дифференцирования вверх	@ +F(454)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Au	Ad	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142		
Рабочая область	W000...W510		
Область удержания бита (Holding bit area)	H000...H510		
Область вспомогательных битов	A000...A958	A448...A958	
Область таймера	T0000...T4094		
Область счетчика	C0000...C4094		
Область DM	D00000...D32766		
Область EM, не содержащая банков	E00000...E32766		
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#00000000...#FFFF FFFF (двоичные)	-	
Регистры данных	-		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15		

**Описание**

По команде +F(454) осуществляется сложение числа с плавающей запятой длиной 32 бита, содержащегося в Ad+1 и Ad, с числом с плавающей запятой длиной 32 бита, содержащимся в Au+1 и Au. Результат записывается слова R+1 и R. (Данные с плавающей запятой должны соответствовать формату IEEE754.)

Au+1	Au	Первое слагаемое. (Данные с плавающей запятой, 32 бита)
Ad+1	Ad	Второе слагаемое. (Данные с плавающей запятой, 32 бита)
R+1	R	Результат. (Данные с плавающей запятой, 32 бита).

Когда абсолютное значение результата превышает максимально возможное значение, которое можно выразить данными с плавающей запятой, флаг переполнения переводится в состояние ON, и результат выводится в виде  $\pm\infty$

Если абсолютное значение результата меньше минимально возможного значения, которое можно выразить данными с плавающей запятой, флаг отрицательного переполнения переводится в состояние ON, и результат выводится в виде «0».

В следующей ниже таблице показаны различные сочетания слагаемых и производимые при этом результаты.

	Первое слагаемое				
Второе слагаемое	0	Числовое значение	$+\infty$	$-\infty$	NaN
0	0	Числовое значение	$+\infty$	$-\infty$	
Числовое значение	Числовое значение	См. примечание 1.	$+\infty$	$-\infty$	
$+\infty$	$+\infty$	$+\infty$	$+\infty$	См. примечание 2.	
$-\infty$	$-\infty$	$-\infty$	См. примечание 2.	$-\infty$	
NaN	См. примечание 2.				

- Примечание:**
1. Результатом может быть ноль (включая отрицательное переполнение), числовое значение, +, или -.
  2. Флаг ошибки переводится в состояние ON и команда не выполняется.

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда данные слагаемых не распознаются как данные с плавающей запятой. Переводится в состояние ON, когда одно из слагаемых не является числом (неопределенность). Переводится в состояние ON, когда выполняется прибавление бесконечности +, или -. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, когда абсолютное значение результата слишком велико, чтобы его можно было выразить числом с плавающей запятой размером в 32 бита.
Флаг отрицательного переполнения	UF	Переводится в состояние ON, когда абсолютное значение результата слишком мало, чтобы его можно было выразить числом с плавающей запятой размером в 32 бита.
Флаг отрицательного значения	N	Переводится в состояние ON, когда результат отрицателен. OFF в других случаях.

#### Меры предосторожности

Данные слагаемых (Au+1 и Au, Ad+1 и Ad) должны быть числами с плавающей запятой, в соответствии с форматом IEEE754.

### 3-15-6 Команда выполнения вычитания двух чисел с плавающей запятой FLOATING-POINT SUBTRACT: -F(455)

#### Назначение

По команде -F(455) осуществляется вычитание одного числа длиной 32 бита с плавающей запятой из другого. Результат записывается в указанные слова.

**Символ релейно-контактной схемы**

-F(455)	
Mi	Mi: Первое слово уменьшаемого
Su	Su: Первое слово вычитаемого
R	R: Первое слово результата

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	-F(455)
	Выполняется один раз для дифференцирования вверх	@ -F(455)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Mi	Su	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142		
Рабочая область	W000...W510		
Область удержания бита (Holding bit area)	H000...H510		
Область вспомогательных битов	A000...A958	A448...A958	
Область таймера	T0000...T4094		
Область счетчика	C0000...C4094		
Область DM	D00000...D32766		
Область EM, не содержащая банков	E00000...E32766		
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#00000000...#FFFF FFFF (двоичные)		
Регистры данных	-		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15		

**Описание**

По команде -F(455) осуществляется вычитание числа с плавающей запятой длиной 32 бита, содержащегося в Su+1 и Su, из числа с плавающей запятой длиной 32 бита, содержащегося в Mi+1 и Mi. Результат записывается слова R+1 и R. (Данные с плавающей запятой должны соответствовать формату IEEE754.)

Mi+1	Mi	Уменьшаемое. (Данные с плавающей запятой, 32 бита).
Su+1	Su	Вычитаемое. (Данные с плавающей запятой, 32 бита)
R+1	R	Результат. (Данные с плавающей запятой, 32 бита).

Когда абсолютное значение результата превышает максимально возможное значение, которое можно выразить данными с плавающей запятой, флаг переполнения переводится в состояние ON, и результат выводится в виде  $\pm\infty$

Если абсолютное значение результата меньше минимально возможного значения, которое можно выразить данными с плавающей запятой, флаг отрицательного переполнения переводится в состояние ON, и результат выводится в виде «0».

В следующей ниже таблице показаны различные сочетания уменьшаемых и вычитаемых и получаемые при этом результаты.

Вычитаемое	Уменьшаемое				
	0	Числовое значение	$+\infty$	$-\infty$	NaN
0	0	Числовое значение	$+\infty$	$-\infty$	
Числовое значение	Числовое значение	См. примечание 1.	$+\infty$	$-\infty$	
$+\infty$	$-\infty$	$-\infty$	См. примечание 2.	$-\infty$	
$-\infty$	$+\infty$	$+\infty$	$+\infty$	См. примечание 2.	
NaN	См. примечание 2.				

- Примечание:**
1. Результатом может быть ноль (включая отрицательные переполнения), числовое значение, + или - .
  2. Флаг ошибки переводится в состояние ON и команда не выполняется.

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда данные уменьшаемого или вычитаемого не распознаются как данные с плавающей запятой. Переводится в состояние ON, когда уменьшаемое или вычитаемое не является числом (неопределенность). Переводится в состояние ON, когда выполняется вычитание - из +. Переводится в состояние ON, когда выполняется вычитание +, из -. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, когда абсолютное значение результата слишком велико, чтобы его можно было выразить числом с плавающей запятой размером в 32 бита.
Флаг отрицательного переполнения	UF	Переводится в состояние ON, когда абсолютное значение результата слишком мало, чтобы его можно было выразить числом с плавающей запятой размером в 32 бита.
Флаг отрицательного значения	N	Переводится в состояние ON, когда результат отрицателен. OFF в других случаях.

#### Меры предосторожности

Данные уменьшаемого и вычитаемого ( $M_{i+1}$  и  $M_i$ ,  $S_{u+1}$  и  $S_u$ ) должны быть числами с плавающей запятой, в соответствии с форматом IEEE754.

### 3-15-7 Команда выполнения умножения двух чисел с плавающей запятой FLOATING-POINT MULTIPLY: \*F(456)

#### Назначение

По команде \*F(456) осуществляется умножение двух чисел длиной 32 бита с плавающей запятой. Результат записывается в указанные слова.

#### Символ релейно-контактной схемы

*F(456)	
Md	Md: Первое слово множимого
Mr	Mr: Первое слово множителя
R	R: Первое слово результата (произведения)

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	*F(456)
	Выполняется один раз для дифференцирования вверх	@ *F(456)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Md	Mr	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142		
Рабочая область	W000...W510		
Область удержания бита (Holding bit area)	H000...H510		
Область вспомогательных битов	A000...A958	A448...A958	
Область таймера	T0000...T4094		
Область счетчика	C0000...C4094		
Область DM	D00000...D32766		
Область EM, не содержащая банков	E00000...E32766		
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#00000000...#FFFF FFFF (двоичные)		
Регистры данных	—		
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15		

**Описание**

По команде \*F(456) осуществляется умножение числа с плавающей запятой длиной 32 бита, содержащегося в Md+1 и Md, на число с плавающей запятой длиной 32 бита, содержащееся в Mr+1 и Mr. Результат записывается в R+1 и R. (Данные с плавающей запятой должны соответствовать формату IEEE754.)

Md+1	Md
------	----

 Множимое. (Данные с плавающей запятой, 32 бита).

⊗ 

Mr+1	Mr
------	----

 Множитель. (Данные с плавающей запятой, 32 бита)

R+1	R
-----	---

 Результат (произведение). (Данные с плавающей запятой, 32 бита).

Когда абсолютное значение результата превышает максимально возможное значение, которое можно выразить данными с плавающей запятой, флаг переполнения переводится в состояние ON, и результат выводится в виде ±∞

Если абсолютное значение результата меньше минимально возможного значения, которое можно выразить данными с плавающей запятой, флаг отрицательного переполнения переводится в состояние ON, и результат выводится в виде «0».

В следующей ниже таблице показаны различные сочетания множимого и множителя и получаемые при этом результаты.

	Множимое				
Множитель	0	Числовое значение	+∞	-∞	NaN
0	0	0	См. примечание 2.	См. примечание 2.	

		Множимое			
Числовое значение	0	См. примечание 1.	$+\infty$	$\pm\infty$	
$+\infty$	См. примечание 2.	$\pm\infty$	$+\infty$	$-\infty$	
$-\infty$	См. примечание 2.	$\pm\infty$	$-\infty$	$+\infty$	
NaN					См. примечание 2.

- Примечание:**
1. Результатом может быть ноль (включая отрицательные переполнения), числовое значение, + или - .
  2. Флаг ошибки переводится в состояние ON и команда не выполняется.

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда данные множимого или множителя не распознаются как данные с плавающей запятой. Переводится в состояние ON, когда множимое или множитель не является числом (неопределенность). Переводится в состояние ON, когда выполняется умножение + и 0. Переводится в состояние ON, когда выполняется умножение -, и 0. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, когда абсолютное значение результата слишком велико, чтобы его можно было выразить числом с плавающей запятой размером в 32 бита.
Флаг отрицательного переполнения	UF	Переводится в состояние ON, когда абсолютное значение результата слишком мало, чтобы его можно было выразить числом с плавающей запятой размером в 32 бита.
Флаг отрицательного значения	N	Переводится в состояние ON, когда результат отрицателен. OFF в других случаях.

#### Меры предосторожности

Данные множимого и множителя (Md+1 и Md, Mr+1 и Mr) должны быть числами с плавающей запятой, в соответствии с форматом IEEE754.

### 3-15-8 Команда выполнения деления двух чисел с плавающей запятой FLOATING-POINT DIVIDE: /F(457)

#### Назначение

По команде /F(457) осуществляется деление одного числа длиной 32 бита с плавающей запятой на другое число с плавающей запятой длиной 32 бита. Результат записывается в указанные слова.

#### Символ релейно-контактной схемы

/F(457)	
Dd	Dd: Первое слово делимого
Dr	Dr: Первое слово делителя
R	R: Первое слово результата (частного)

#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	/F(457)
	Выполняется один раз для дифференцирования вверх	@ /F(457)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

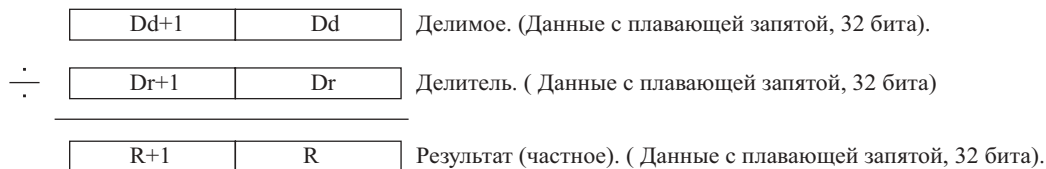
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	Dd	Dr	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142		
Рабочая область	W000...W510		
Область удержания бита (Holding bit area)	H000...H510		
Область вспомогательных битов	A000...A958	A448...A958	
Область таймера	T0000...T4094		
Область счетчика	C0000...C4094		
Область DM	D00000...D32766		
Область EM, не содержащая банков	E00000...E32766		
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#00000000...#FFFF FFFF (двоичные)	-	
Регистры данных	-		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,(- -)IR0...,(- -)IR15		

**Описание**

По команде /F(457) осуществляется деление числа с плавающей запятой длиной 32 бита, содержащегося в Dd+1 и Dd, на число с плавающей запятой длиной 32 бита, содержащееся в Dr+1 и Dr. Результат записывается слова R+1 и R. (Данные с плавающей запятой должны соответствовать формату IEEE754.)



Когда абсолютное значение результата превышает максимально возможное значение, которое можно выразить данными с плавающей запятой, флаг переполнения переводится в состояние ON, и результат выводится в виде ±∞.

Если абсолютное значение результата меньше минимально возможного значения, которое можно выразить данными с плавающей запятой, флаг отрицательного переполнения переводится в состояние ON, и результат выводится в виде «0».

В следующей ниже таблице показаны различные сочетания делимого и делителя и получаемые при этом результаты.

	Делимое				
Делитель	0	Числовое значение	+∞	-∞	NaN
0	См. примечание 3.	±∞	+∞	-∞	
Числовое значение	0	См. примечание 1.	±∞	±∞	
+∞	0	См. примечание 2.	См. примечание 3.	См. примечание 3.	
-∞	0	См. примечание 2.	См. примечание 3.	См. примечание 3.	
NaN					См. примечание 2.

- Примечание:**
1. Результатом может быть ноль (включая отрицательные переполнения), числовое значение, + или - .
  2. При определении отрицательного переполнения в качестве результата выводится 0.
  3. Флаг ошибки переводится в состояние ON и команда не выполняется.



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда данные делимого или делителя не распознаются как данные с плавающей запятой. Переводится в состояние ON, когда множимое или множитель не является числом (неопределенность). Переводится в состояние ON, когда множимое и множитель одновременно равны нулю. Переводится в состояние ON, когда множимое и множитель одновременно равны + или - . OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, когда абсолютное значение результата слишком велико, чтобы его можно было выразить числом с плавающей запятой размером в 32 бита.
Флаг отрицательного переполнения	UF	Переводится в состояние ON, когда абсолютное значение результата слишком мало, чтобы его можно было выразить числом с плавающей запятой размером в 32 бита.
Флаг отрицательного значения	N	Переводится в состояние ON, когда результат отрицателен. OFF в других случаях.

**Меры предосторожности**

Данные делимого и делителя (Dd+1 и Dd, Dr+1 и Dr) должны быть числами с плавающей запятой, в соответствии с форматом IEEE754.

**3-15-9 Команда преобразования градусов в радианы DEGREES TO RADIANS: RAD(458)****Назначение**

По команде RAD(458) производится преобразование одного числа длиной 32 бита с плавающей запятой из градусов в радианы. Результат записывается в указанные слова.

**Символ релейно-контактной схемы**

—	RAD(458)	
	S	S: Первое исходное слово
	R	R: Первое слово результата

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	RAD(458)
	Выполняется один раз для дифференцирования вверх	@ RAD(458)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

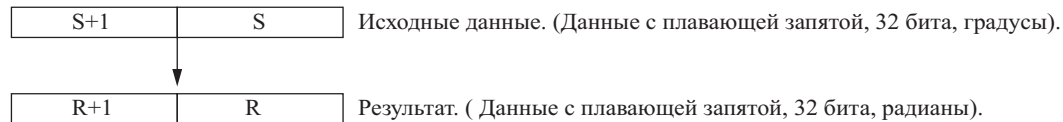
**Спецификации операндов**

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	

Область	S	R
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFF (двоичные)	–
Регистры данных	–	
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(++) ,(- -)IR0...(- -)IR15	

**Описание**

По команде RAD(458) производится преобразование из градусов в радианы числа с плавающей запятой длиной 32 бита, содержащегося в S+1 и S. Результат записывается слова R+1 и R. (Данные с плавающей запятой должны соответствовать формату IEEE754.)



Преобразование градусов в радианы производится по формуле:

$$\text{Градусы} \times \pi / 180 = \text{радианы}$$

Когда абсолютное значение результата превышает максимально возможное значение, которое можно выразить данными с плавающей запятой, флаг переполнения переводится в состояние ON, и результат выводится в виде  $\pm\infty$ .

Если абсолютное значение результата меньше минимально возможного значения, которое можно выразить данными с плавающей запятой, флаг отрицательного переполнения переводится в состояние ON, и результат выводится в виде «0».

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда исходные данные не распознаются как данные с плавающей запятой. Переводится в состояние ON, когда исходные данные не являются числом (неопределенность NaN). OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, когда абсолютное значение результата слишком велико, чтобы его можно было выразить числом с плавающей запятой размером в 32 бита.
Флаг отрицательного переполнения	UF	Переводится в состояние ON, когда абсолютное значение результата слишком мало, чтобы его можно было выразить числом с плавающей запятой размером в 32 бита.
Флаг отрицательного значения	N	Переводится в состояние ON, когда результат отрицателен. OFF в других случаях.

**Меры предосторожности**

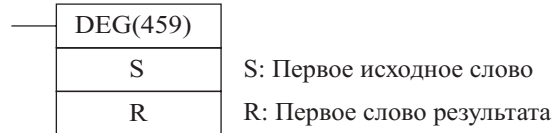
Исходные данные в S+1 и S должны быть числами с плавающей запятой, в соответствии с форматом IEEE754.

3-15-10 Команда преобразования радианов в градусы RADIANS TO DEGREES: DEG(459)

**Назначение**

По команде DEG(459) производится преобразование одного числа длиной 32 бита с плавающей запятой из радианов в градусы. Результат записывается в указанные слова.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	DEG(459)
	Выполняется один раз для дифференцирования вверх	@ DEG(459)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

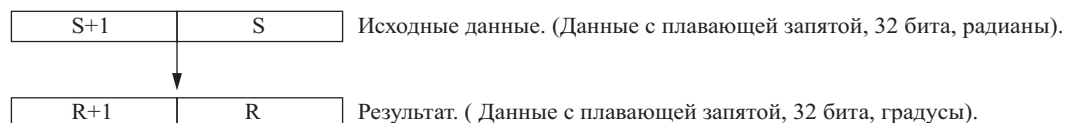
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFF (двоичные)	—
Регистры данных	—	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-(-)IR15	

**Описание**

По команде DEG(459) производится преобразование числа с плавающей запятой длиной 32 бита, содержащегося в S+1 и S, из радианов в градусы. Результат записывается слова R+1 и R. (Данные с плавающей запятой должны соответствовать формату IEEE754.)



Преобразование радианов в градусы производится по формуле:

$$\text{радианы} \times 180 / \pi = \text{градусы}$$

Когда абсолютное значение результата превышает максимально возможное значение, которое можно выразить данными с плавающей запятой, флаг переполнения переводится в состояние ON, и результат выводится в виде  $\pm\infty$ .

Если абсолютное значение результата меньше минимально возможного значения, которое можно выразить данными с плавающей запятой, флаг отрицательного переполнения переводится в состояние ON, и результат выводится в виде «0».

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда исходные данные не распознаются как данные с плавающей запятой. Переводится в состояние ON, когда исходные данные не являются числом (неопределенность NaN). OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, когда абсолютное значение результата слишком велико, чтобы его можно было выразить числом с плавающей запятой размером в 32 бита.
Флаг отрицательного переполнения	UF	Переводится в состояние ON, когда абсолютное значение результата слишком мало, чтобы его можно было выразить числом с плавающей запятой размером в 32 бита.
Флаг отрицательного значения	N	Переводится в состояние ON, когда результат отрицателен. OFF в других случаях.

#### Меры предосторожности

Исходные данные в S+1 и S должны быть числами с плавающей запятой, в соответствии с форматом IEEE754.

### 3-15-11 Команда вычисления синуса SINE: SIN(460)

#### Назначение

По команде SIN(460) производится вычисление синуса числа длиной 32 бита с плавающей запятой (в радианах). Результат записывается в указанные слова.

#### Символ релейно-контактной схемы

SIN(460)	
S	S: Первое исходное слово
R	R: Первое слово результата

#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	SIN(460)
	Выполняется один раз для дифференцирования вверх	@ SIN(460)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Спецификации операндов

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	

Область	S	R
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFFF (двоичные)	–
Регистры данных	–	
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...(- -)IR15	

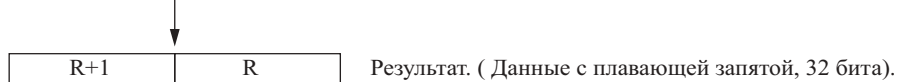
**Описание**

По команде SIN(460) производится вычисление синуса угла (в радианах), выраженного числом длиной 32 бита с плавающей запятой. Исходные данные содержатся в S+1 и S. Результат записывается в слова R+1 и R. (Данные с плавающей запятой должны соответствовать формату IEEE754.)

SIN ( 

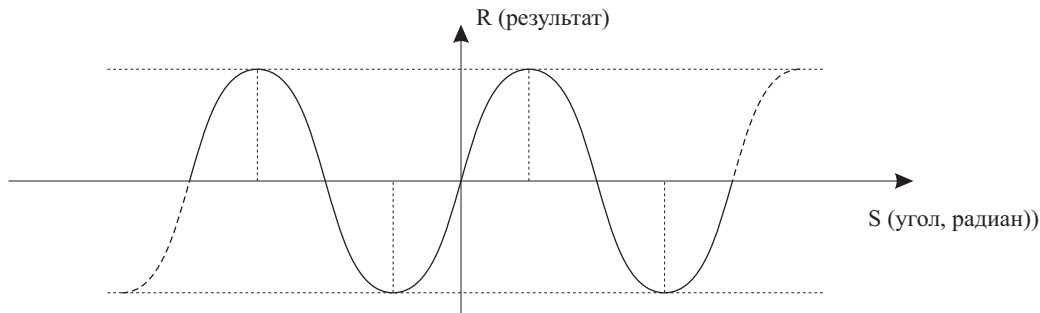
S+1	S
-----	---

 ) Исходные данные. (Данные с плавающей запятой, 32 бита).



В словах S+1 и S указывайте угол в радианах (от -65535 до 65535). Если абсолютное значение задаваемого угла превышает значение, равное 65535, определяется ошибка и команда не выполняется. Для ознакомления с процессом преобразования градусов в радианы обратитесь к разделу 3-15-9 «Команда преобразования градусов в радианы DEGREES TO RADIANS: RAD(458)».

На следующем ниже рисунке представлена зависимость величины результата от градусов.



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда исходные данные не являются числом (неопределенность NaN). Переводится в состояние ON, когда абсолютное значение исходных данных превышает 65535. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг переполнения	OF	OFF
Флаг отрицательного переполнения	UF	OFF
Флаг отрицательного значения	N	Переводится в состояние ON, когда результат отрицателен. OFF в других случаях.

**Меры предосторожности**

Исходные данные в S+1 и S должны быть числами с плавающей запятой, в соответствии с форматом IEEE754.

**3-15-12 Команда вычисления косинуса COSINE: COS(461)****Назначение**

По команде COS(461) производится вычисление косинуса числа длиной 32 бита с плавающей запятой (в радианах). Результат записывается в указанные слова.

**Символ релейно-контактной схемы**

—	COS(461)	
	S	S: Первое исходное слово
	R	R: Первое слово результата

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	COS(461)
	Выполняется один раз для дифференцирования вверх	@ COS(461)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

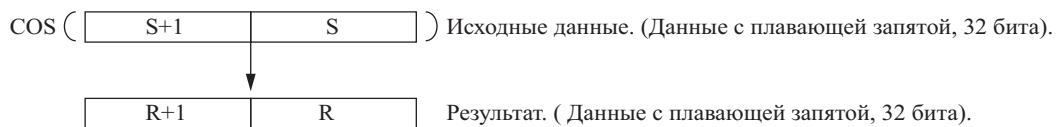
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFF (двоичные)	—
Регистры данных	—	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(++) ,(- -)IR0...(- -)IR15	

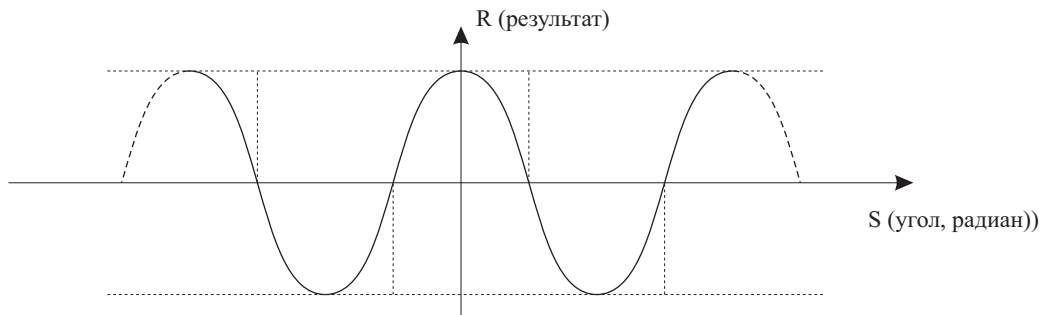
**Описание**

По команде COS(461) производится вычисление косинуса угла (в радианах), выраженного числом длиной 32 бита с плавающей запятой. Исходные данные содержатся в S+1 и S. Результат записывается в слова R+1 и R. (Данные с плавающей запятой должны соответствовать формату IEEE754.)



В словах S+1 и S указывайте угол в радианах (от -65535 до 65535). Если абсолютное значение задаваемого угла превышает значение, равное 65535, определяется ошибка и команда не выполняется. Для ознакомления с процессом преобразования градусов в радианы обратитесь к разделу 3-15-9 «Команда преобразования градусов в радианы *DEGREES TO RADIANS: RAD(458)*».

На следующем ниже рисунке представлена зависимость величины результата от градусов.



#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда исходные данные не являются числом (неопределенность NaN). Переводится в состояние ON, когда абсолютное значение исходных данных превышает 65535. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг переполнения	OF	OFF
Флаг отрицательного переполнения	UF	OFF
Флаг отрицательного значения	N	Переводится в состояние ON, когда результат отрицателен. OFF в других случаях.

#### Меры предосторожности

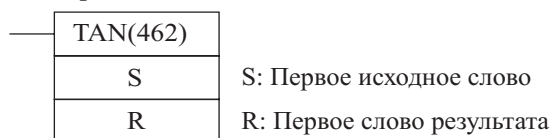
Исходные данные в S+1 и S должны быть числами с плавающей запятой, в соответствии с форматом IEEE754.

#### 3-15-13 Команда вычисления тангенса TANGENT: TAN(462)

##### Назначение

По команде TAN(462) производится вычисление тангенса числа длиной 32 бита с плавающей запятой (в радианах). Результат записывается в указанные слова.

##### Символ релейно-контактной схемы



##### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	TAN(462)
	Выполняется один раз для дифференцирования вверх	@ TAN(462)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

## Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

## Спецификации операндов

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFF (двоичные)	–
Регистры данных	–	
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15	

## Описание

По команде TAN(462) производится вычисление тангенса угла (в радианах), выраженного числом длиной 32 бита с плавающей запятой. Исходные данные содержатся в S+1 и S. Результат записывается в слова R+1 и R. (Данные с плавающей запятой должны соответствовать формату IEEE754.)

TAN ( 

S+1	S
-----	---

 ) Исходные данные. (Данные с плавающей запятой, 32 бита).

↓

R+1	R
-----	---

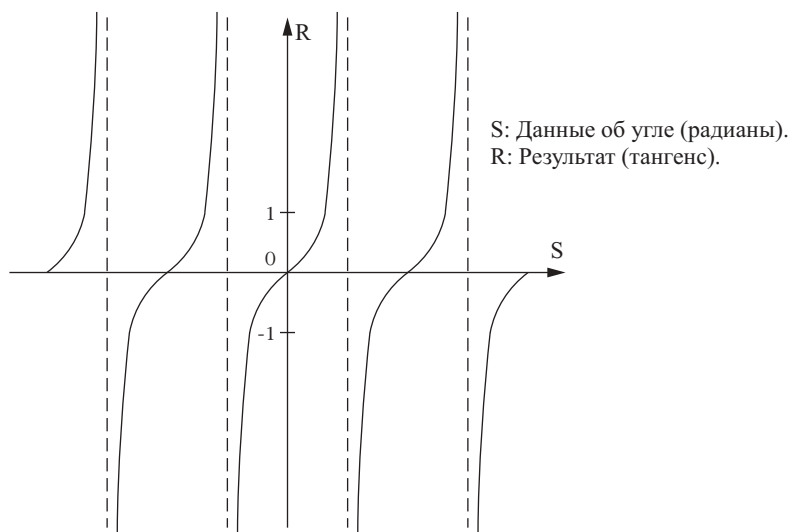
 Результат. (Данные с плавающей запятой, 32 бита).

В словах S+1 и S указывайте угол в радианах (от –65535 до 65535). Если абсолютное значение задаваемого угла превышает значение, равное 65535, определяется ошибка и команда не выполняется. Для ознакомления с процессом преобразования градусов в радианы обратитесь к разделу 3-15-9 «Команда преобразования градусов в радианы DEGREES TO RADIANS: RAD(458)».

Если абсолютное значение результата превышает максимально возможное значение, которое можно выразить данными с плавающей запятой, флаг переполнения переводится в состояние ON, и результат выводится в виде  $\pm\infty$ .

На следующем ниже рисунке представлена зависимость величины результата от градусов.





**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда исходные данные не являются числом (неопределенность NaN). Переводится в состояние ON, когда абсолютное значение исходных данных превышает 65535. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг переполнения	OF	OFF
Флаг отрицательного переполнения	UF	OFF
Флаг отрицательного значения	N	Переводится в состояние ON, когда результат отрицателен. OFF в других случаях.

**Меры предосторожности**

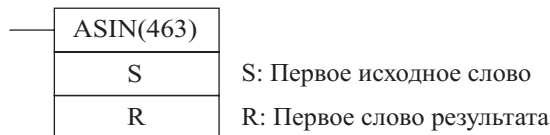
Исходные данные в S+1 и S должны быть числами с плавающей запятой, в соответствии с форматом IEEE754.

**3-15-14 Команда вычисления арксинуса ARC SINE: ASIN(463)**

**Назначение**

По команде ASIN(463) осуществляется вычисление арксинуса числа длиной 32 бита с плавающей запятой. (Арксинус является обратной функцией по отношению к синусу. Арксинус позволяет вычислять угол из заданного значения синуса в интервале от -1 до 1). Результат записывается в указанные слова.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	ASIN(463)
	Выполняется один раз для дифференцирования вверх	@ ASIN(463)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

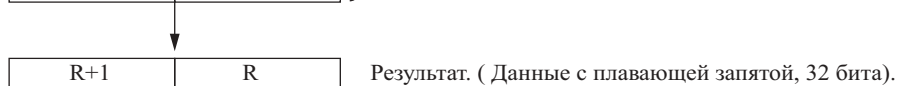
**Спецификации операндов**

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFF (двоичные)	-
Регистры данных	-	
Индексные регистры	-	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15	

**Описание**

По команде ASIN(463) производится вычисление угла (в радианах) из заданного значения синуса, выраженного числом с плавающей запятой длиной 32 бита. Исходное число содержится в S+1 и S. Результат записывается в слова R+1 и R. (Данные с плавающей запятой должны соответствовать формату IEEE754.)

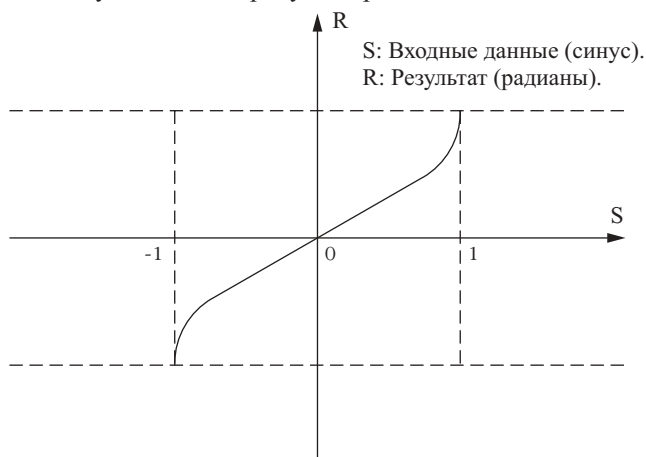
$\text{SIN}^{-1} \left( \begin{array}{|c|c|} \hline \text{S+1} & \text{S} \\ \hline \end{array} \right)$  Исходные данные. (Данные с плавающей запятой, 32 бита).



Диапазон значений исходного числа от -1 до 1. Если абсолютное значение задаваемого угла превышает значение, равное 1.0, определяется ошибка и команда не выполняется.

Результат – величина угла, выраженная в радианах, записывается в слова R+1 и R. Диапазон изменения величины результата – от  $-\pi/2$  до  $\pi/2$ .

На следующем ниже рисунке представлена зависимость величины результата от исходных данных.



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда исходные данные не распознаются как данные с плавающей запятой. Переводится в состояние ON, когда исходные данные не являются числом (неопределенность NaN). Переводится в состояние ON, когда абсолютное значение исходных данных превышает 1.0. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг переполнения	OF	OFF
Флаг отрицательного переполнения	UF	OFF
Флаг отрицательного значения	N	Переводится в состояние ON, когда результат отрицателен. OFF в других случаях.

**Меры предосторожности**

Исходные данные в S+1 и S должны быть числами с плавающей запятой, в соответствии с форматом IEEE754.

**3-15-15 Команда вычисления арккосинуса ARC COSINE: ACOS(464)****Назначение**

По команде ACOS(464) производится вычисление арккосинуса числа длиной 32 бита с плавающей запятой. (Арккосинус является обратной функцией по отношению к косинусу. Арккосинус позволяет вычислять угол из заданного значения косинуса в интервале от  $-1$  до  $1$ ). Результат записывается в указанные слова.

**Символ релейно-контактной схемы**

—	ACOS(464)	
	S	S: Первое исходное слово
	R	R: Первое слово результата

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	ACOS(464)
	Выполняется один раз для дифференцирования вверх	@ ACOS(464)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

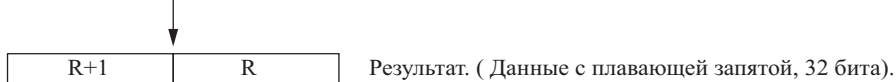
Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	

Область	S	R
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFF (двоичные)	—
Регистры данных	—	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...IR15(++) ,(- -)IR0...(- -)IR15	

**Описание**

По команде ACOS(464) производится вычисление угла (в радианах) из заданного значения косинуса, выраженного числом с плавающей запятой длиной 32 бита. Исходное число содержится в S+1 и S. Результат записывается в слова R+1 и R. (Данные с плавающей запятой должны соответствовать формату IEEE754.)

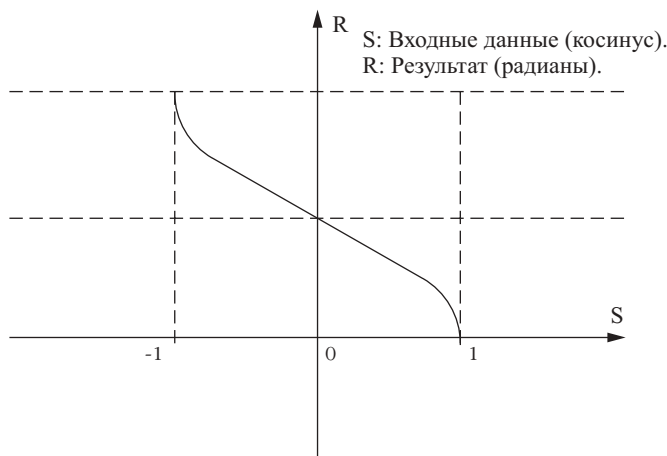
$\text{COS}^{-1} \left( \begin{array}{|c|c|} \hline \text{S+1} & \text{S} \\ \hline \end{array} \right)$  Исходные данные. (Данные с плавающей запятой, 32 бита).



Диапазон значений исходного числа от -1 до 1. Если абсолютное значение задаваемого угла превышает значение, равное 1.0, определяется ошибка и команда не выполняется.

Результат – величина угла, выраженная в радианах, записывается в слова R+1 и R. Диапазон изменения величины результата – от 0 до π.

На следующем ниже рисунке представлена зависимость величины результата от исходных данных.



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда исходные данные не распознаются как данные с плавающей запятой. Переводится в состояние ON, когда исходные данные не являются числом (неопределенность NaN). Переводится в состояние ON, когда абсолютное значение исходных данных превышает 1.0. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг переполнения	OF	OFF
Флаг отрицательного переполнения	UF	OFF

Наименование	Обозначение	Функционирование
Флаг отрицательного значения	N	Переводится в состояние ON, когда результат отрицателен. OFF в других случаях.

**Меры предосторожности**

Исходные данные в S+1 и S должны быть числами с плавающей запятой, в соответствии с форматом IEEE754.

**3-15-16 Команда вычисления арктангенса ARC TANGENT: ATAN(465)****Назначение**

По команде ATAN(465) производится вычисление арктангенса числа длиной 32 бита с плавающей запятой. (Арктангенс является обратной функцией по отношению к тангенсу. Арктангенс позволяет вычислять угол из заданного значения тангенса). Результат записывается в указанные слова.

**Символ релейно-контактной схемы**

—	ATAN(465)	
	S	S: Первое исходное слово
	R	R: Первое слово результата

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	ATAN(465)
	Выполняется один раз для дифференцирования вверх	@ ATAN(465)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFF (двоичные)	—
Регистры данных	—	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15	

**Описание**

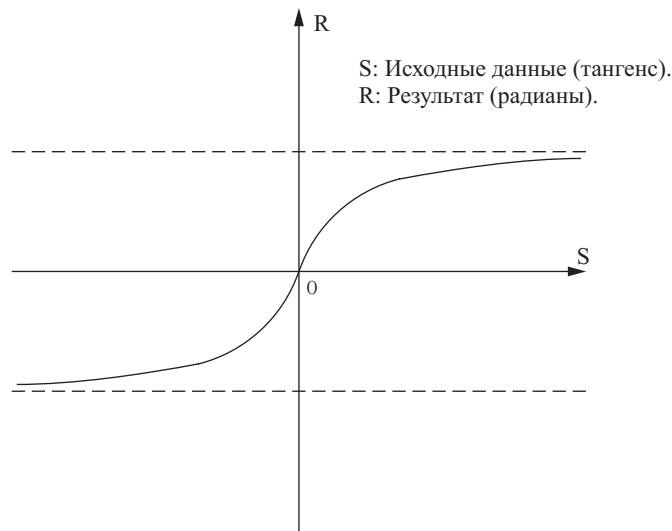
По команде ATAN(465) производится вычисление угла (в радианах), из заданного значения тангенса, выраженного числом с плавающей запятой длиной 32 бита. Исходные данные содержатся в S+1 и S. Результат записывается в слова R+1 и R. (Данные с плавающей запятой должны соответствовать формату IEEE754.)

$TAN^{-1} \left( \begin{array}{|c|c|} \hline S+1 & S \\ \hline \end{array} \right)$  Исходные данные. (Данные с плавающей запятой, 32 бита).

$\begin{array}{|c|c|} \hline R+1 & R \\ \hline \end{array}$  Результат. (Данные с плавающей запятой, 32 бита).

Результат – величина угла, выраженная в радианах, записывается в слова R+1 и R. Диапазон изменения величины результата – от  $-\pi/2$  до  $\pi/2$ .

На следующем ниже рисунке представлена зависимость величины результата от градусов.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда исходные данные не распознаются как данные с плавающей запятой. Переводится в состояние ON, когда исходные данные не являются числом (неопределенность NaN). OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг переполнения	OF	OFF
Флаг отрицательного переполнения	UF	OFF
Флаг отрицательного значения	N	Переводится в состояние ON, когда результат является отрицательным числом. OFF в других случаях.

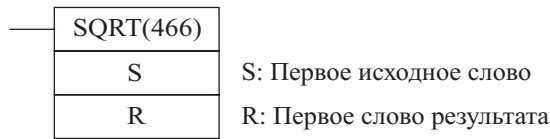
**Меры предосторожности**

Исходные данные в S+1 и S должны быть числами с плавающей запятой, в соответствии с форматом IEEE754.

**3-15-17 Команда вычисления квадратного корня SQUARE ROOT: SQRT(466)****Назначение**

По команде SQRT(466) производится вычисление квадратного корня из числа с плавающей запятой длиной 32 бита. Результат записывается в указанные слова.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	SQRT(466)
	Выполняется один раз для дифференцирования вверх	@ SQRT(466)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

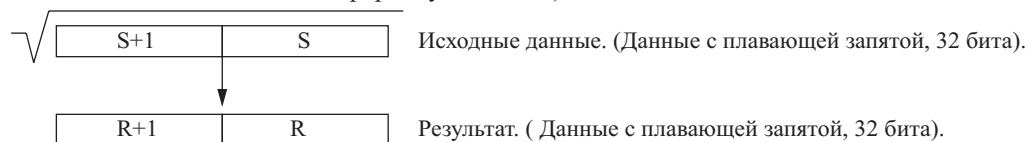
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFF (двоичные)	—
Регистры данных	—	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-(-)IR15	

**Описание**

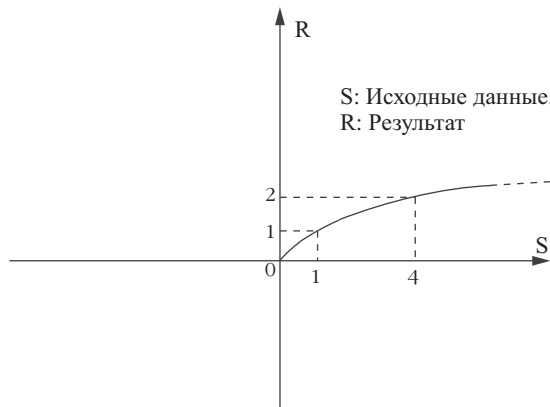
По команде SQRT(466) производится вычисление квадратного корня из числа с плавающей запятой длиной 32 бита, находящегося в S+1 и S. Результат записывается в слова R+1 и R. (Данные с плавающей запятой должны соответствовать формату IEEE754.)



Исходные данные должны быть положительным числом, в противном случае определяется ошибка, и команда не выполняется.

Если абсолютное значение результата превышает максимально возможное значение, которое можно выразить данными с плавающей запятой, флаг переполнения переводится в состояние ON, и результат выводится в виде ±∞.

На следующем ниже рисунке представлена зависимость величины результата от исходных данных.



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда исходные данные не распознаются как данные с плавающей запятой. Переводится в состояние ON, когда исходные данные являются отрицательным числом. Переводится в состояние ON, когда исходные данные не являются числом (неопределенность NaN). OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, если абсолютное значение результата слишком велико, чтобы его можно было выразить числом с плавающей запятой длиной 32 бита.
Флаг отрицательного переполнения	UF	OFF
Флаг отрицательного значения	N	OFF.

**Меры предосторожности**

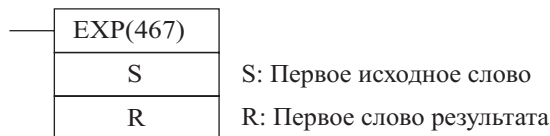
Исходные данные в S+1 и S должны быть числами с плавающей запятой, в соответствии с форматом IEEE754.

3-15-18 Команда вычисления натуральной экспоненты EXPONENT: EXP(467)

**Назначение**

По команде EXP(467) производится вычисление натуральной (по основанию e) экспоненты числа с плавающей запятой длиной 32 бита. Результат записывается в указанные слова.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	EXP(467)
	Выполняется один раз для дифференцирования вверх	@ EXP(467)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

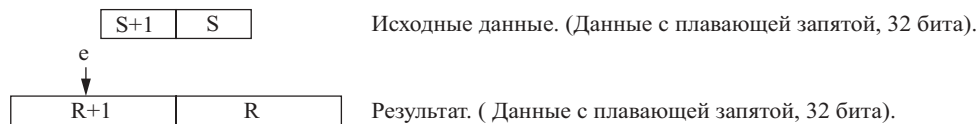


**Спецификации операндов**

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFF (двоичные)	–
Регистры данных	–	
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,(- -)IR0...,(- -)IR15	

**Описание**

По команде EXP(467) производится вычисление натуральной (по основанию e) показательной функции числа с плавающей запятой длиной 32 бита, находящегося в S+1 и S. Другими словами вычисляется  $e^x$  (x=исходные данные). Результат записывается в слова R+1 и R. (Данные с плавающей запятой должны соответствовать формату IEEE754.)

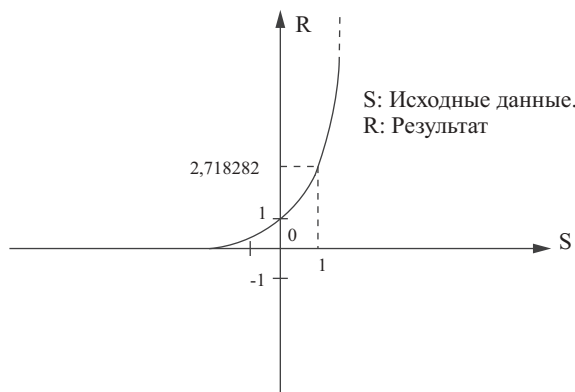


Если абсолютное значение результата превышает максимально возможное значение, которое можно выразить данными с плавающей запятой, флаг переполнения переводится в состояние ON, и результат выводится в виде  $\pm\infty$ .

Если абсолютное значение результата меньше минимально возможного значения, которое можно выразить данными с плавающей запятой, флаг отрицательного переполнения переводится в состояние ON, и результат выводится в виде «0».

**Примечание:** Величина константы e равна 2.718282.

На следующем ниже рисунке представлена зависимость величины результата от исходных данных.



**Флаги**

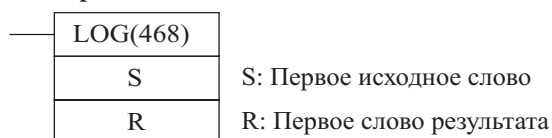
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда исходные данные не распознаются как данные с плавающей запятой. Переводится в состояние ON, когда исходные данные не являются числом (неопределенность NaN). OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, если абсолютное значение результата слишком велико, чтобы его можно было выразить числом с плавающей запятой длиной 32 бита.
Флаг отрицательного переполнения	UF	Переводится в состояние ON, если абсолютное значение результата слишком мало, чтобы его можно было выразить числом с плавающей запятой длиной 32 бита.
Флаг отрицательного значения	N	OFF.

**Меры предосторожности**

Исходные данные в S+1 и S должны быть числами с плавающей запятой, в соответствии с форматом IEEE754.

**3-15-19 Команда вычисления логарифма LOGARITHM: LOG(468)****Назначение**

По команде LOG(468) производится вычисление натурального логарифма (по основанию e) числа с плавающей запятой длиной 32 бита. Результат записывается в указанные слова.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	LOG(468)
	Выполняется один раз для дифференцирования вверх	@ LOG(468)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

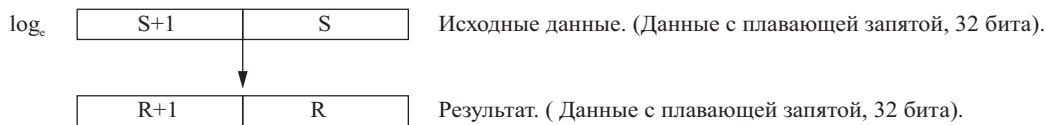
**Спецификации операндов**

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	

Область	S	R
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFF (двоичные)	–
Регистры данных	–	
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)... ,IR15(++) ,(- -)IR0...,(- -)IR15	

**Описание**

По команде LOG(468) производится вычисление натурального логарифма (по основанию e) числа с плавающей запятой длиной 32 бита, находящегося в S+1 и S. Результат записывается в слова R+1 и R.

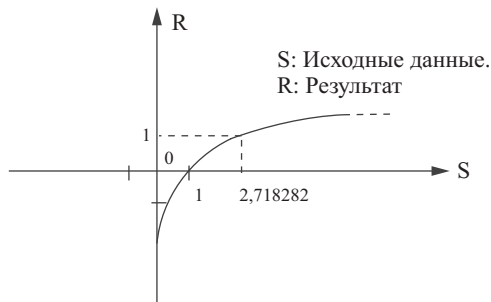


Исходные данные должны быть положительным числом, в противном случае определяется ошибка, и команда не выполняется.

Если абсолютное значение результата превышает максимально возможное значение, которое можно выразить данными с плавающей запятой, флаг переполнения переводится в состояние ON, и результат выводится в виде ±∞.

**Примечание:** Величина константы e равна 2.718282.

На следующем ниже рисунке представлена зависимость величины результата от исходных данных.



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда исходные данные не распознаются как данные с плавающей запятой. Переводится в состояние ON, когда исходные данные являются отрицательной величиной. Переводится в состояние ON, когда исходные данные не являются числом (неопределенность NaN). OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, если абсолютное значение результата слишком велико, чтобы его можно было выразить числом с плавающей запятой длиной 32 бита.
Флаг отрицательного переполнения	UF	OFF
Флаг отрицательного значения	N	Переводится в состояние ON, когда результат является отрицательной величиной. OFF в других случаях.

**Меры предосторожности**

Исходные данные в S+1 и S должны быть числами с плавающей запятой, в соответствии с форматом IEEE754.

**3-15-20 Команда возведения в степень EXPONENTIAL POWER: PWR(840)****Назначение**

По команде PWR(840) производится возведение в степень числа с плавающей запятой длиной 32 бита, при этом показателем степени является другое число с плавающей запятой длиной 32 бита. Результат записывается в указанные слова.

**Символ релейно-контактной схемы**

—	PWR(840)	
	B	B: Первое исходное слово
	E	E: Первое слово показателя степени
	R	R: Первое слово результата

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	PWR(840)
	Выполняется один раз для дифференцирования вверх	@ PWR(840)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

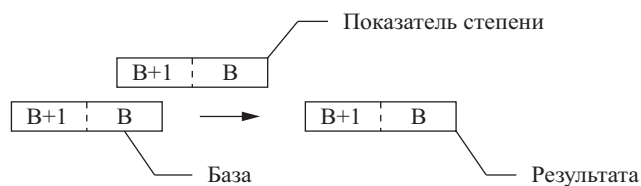
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	S	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W510	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	#00000000...#FFFFFFF (двоичные)	—
Регистры данных	—	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15	

**Описание**

По команде PWR(840) производится возведение в степень числа с плавающей запятой длиной 32 бита, находящегося в B+1 и B. Показателем степени является другое число с плавающей запятой длиной 32 бита, находящееся в E+1 и E. Другими словами производится вычисление по формуле  $X^Y$  ( $X=B+1$ ,  $B$ ,  $a$   $Y=E+1$ ,  $E$ ).



Например, если слова базы (B+1, B) содержат число 3.1, а слова показателя степени (E+1, E) содержат число 3, результат равен:

$$3.1^3 = 29.791$$

Если абсолютное значение результата превышает максимально возможное значение, которое можно выразить данными с плавающей запятой, флаг переполнения переводится в состояние ON, и результат выводится в виде  $\pm\infty$ .

Если абсолютное значение результата меньше минимально возможного значения, которое можно выразить данными с плавающей запятой, флаг отрицательного переполнения переводится в состояние ON.

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда исходные данные (база B+1, B или показатель степени E+1, E) не распознаются как данные с плавающей запятой. Переводится в состояние ON, когда исходные данные (база B+1, B или показатель степени E+1, E) не являются числом (неопределенность NaN). Переводится в состояние ON, когда база (B+1, B) равна нулю, а показатель степени (E+1, E) меньше нуля (деление на 0). Переводится в состояние ON, когда база (B+1, B) является отрицательным числом, а показатель степени (E+1, E) не целое число (извлечение квадратного корня из отрицательного числа). OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда показатель степени и мантисса результата равны нулю. OFF в других случаях.
Флаг переполнения	OF	Переводится в состояние ON, если абсолютное значение результата слишком велико, чтобы его можно было выразить числом с плавающей запятой длиной 32 бита.
Флаг отрицательного переполнения	UF	Переводится в состояние ON, если абсолютное значение результата слишком мало, чтобы его можно было выразить числом с плавающей запятой длиной 32 бита.
Флаг отрицательного значения	N	Переводится в состояние ON, когда результат является отрицательным числом. OFF.

#### Меры предосторожности

Исходные данные - база (B+1, B) и показатель степени (E+1 и E), должны быть числами с плавающей запятой, в соответствии с форматом IEEE754.

## 3-16 Команды для обработки табличных данных

В настоящем разделе приводится описание команд, используемых для обработки табличных данных, стеков и других данных, обладающих диапазонами значений.

Команда	Мнемоническое изображение	Код функции	Страница
SET STACK	SSET	630	458
PUSH ONTO STACK	PUSH	632	460
FIRST IN FIRST OUT	FIFO	633	463
LAST IN FIRST OUT	LIFO	634	465
DIMENSION RECORD TABLE	DIM	631	468
SET RECORD LOCATION	SETR	635	470
GET RECORD NUMBER	GETR	636	472
DATA SEARCH	SRCH	181	474
SWAP BYTES	SWAP	637	477
FIND MAXIMUM	MAX	182	479
FIND MINIMUM	MIN	183	482
SUM	SUM	184	485
FRAME CHECKSUM	FCS	180	488

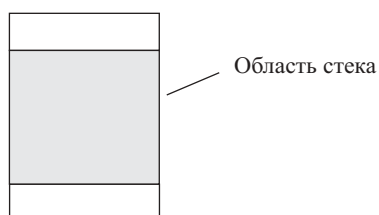
Все перечисленные выше команды определяют или обрабатывают группы слов. Группа слов в стеке определяется командой SSET(630), группа слов в таблице записей определяется командой DIM(631), а группа слов, используемая в команде, обрабатывающей диапазон данных, определяется независимо в каждой из команд.

Группа	Назначение	Команды
Стек	Обрабатывает таблицы данных FIFO (первым вводится, первым выводится) LIFO (последним вводится, первым выводится)	SSET(630), PUSH(632), FIFO(633), LIFO(634).
Таблица записей	Обрабатывает таблицы данных, составленных из записей (размер записи определяется пользователем).	DIM(631), SETR(635), GETR(636).
Диапазон	Обрабатывает диапазон слов для нахождения таких величин как контрольная сумма, определенное значение, максимальное значение или минимальное значение в диапазоне.	FCS(180), SRCH(181), MAX(182), MIN(183), SUM(184), SWAP(637)

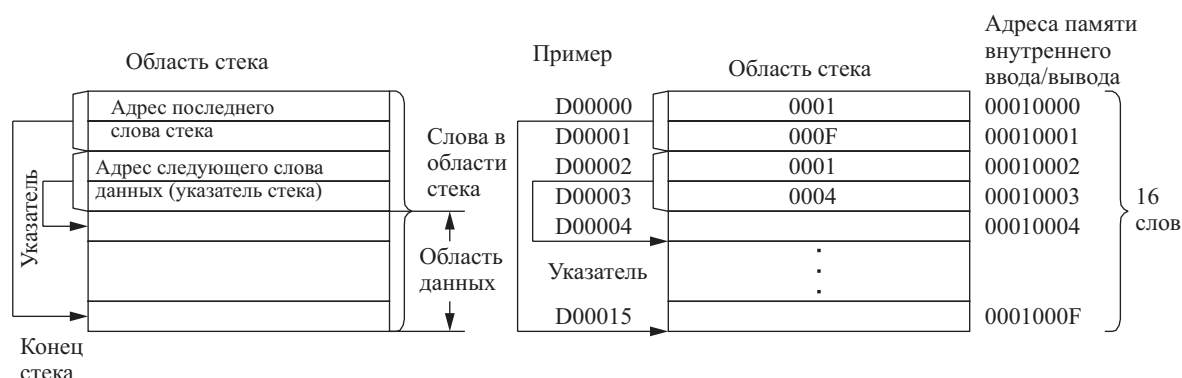
**Команды обработки стеков**

Команды обработки стеков предназначены для выполнения операций со специальными таблицами данных, называемыми стеками. Два первых слова стека содержат адрес последнего слова стека, а два следующих слова содержат указатель стека (адрес слова, которое будет перезаписано при выполнении следующей команды PUSH(632)).

Память  
ввода/вывода



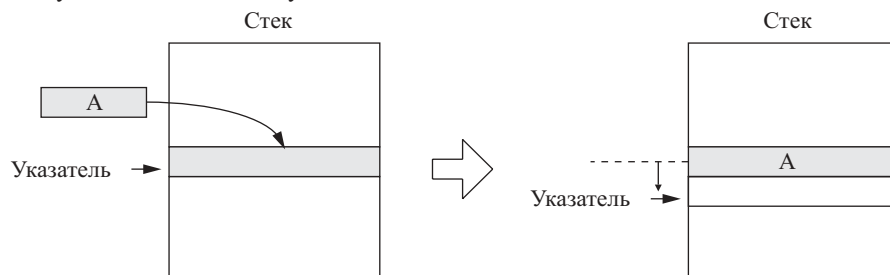
На следующем ниже рисунке показана базовая структура стека.



Следующие ниже команды определяют область стека или выполняют операции с областями стека. Обычно, по команде PUSH(632) осуществляется сохранение данных в следующем доступном слове данных, находящемся в стеке. Команды FIFO(633) и LIFO(634) осуществляют чтение данных в стеке. По команде FIFO(633) производится чтение слова, которое было записано первым, а по команде LIFO(634) – чтение слова, которое было записано последним.

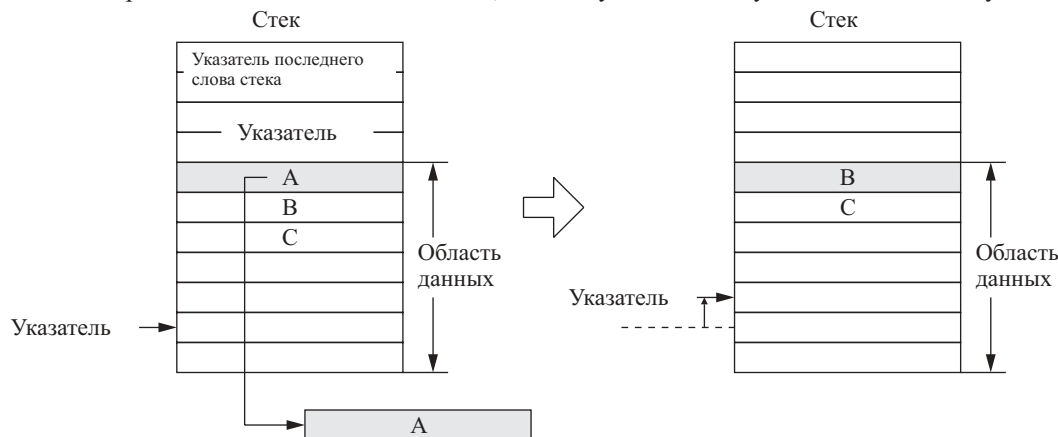
#### **PUSH(632)**

По команде PUSH(632) производится запись данных в адрес, обозначенный указателем стека, и увеличение указателя на единицу.



#### **FIFO(633)**

По команде FIFO(633) производится чтение слова, которое было записано в стек первым, остающиеся данные передвигаются вниз на одно слово, а также уменьшается указатель на единицу.

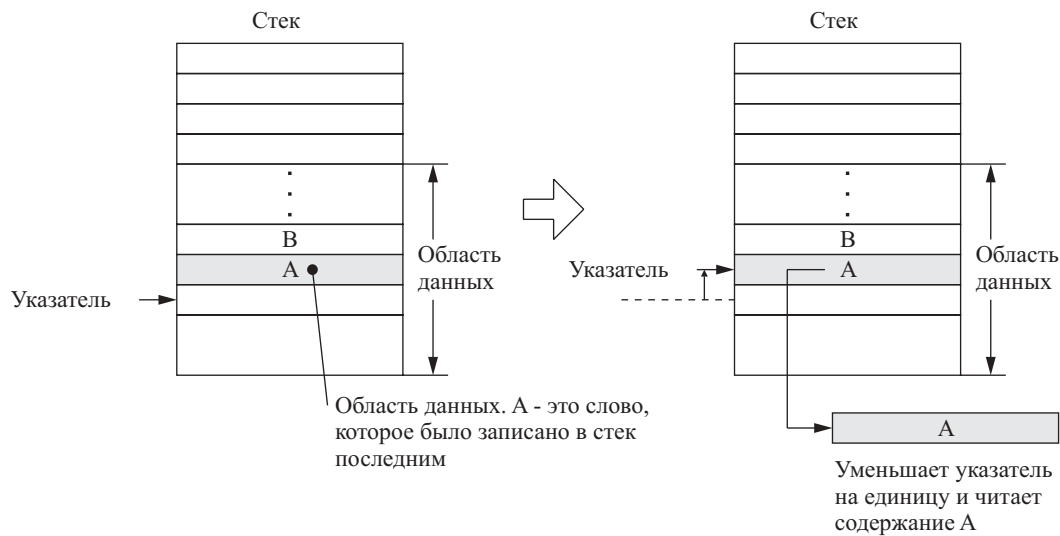


Осуществляет чтение содержания A, уменьшает указатель на единицу, удаляет содержание A, и перемещает остающиеся данные вниз (на одно слово).

Осуществляет чтение содержания A, уменьшает указатель на единицу, удаляет содержание A, и перемещает остающиеся данные вниз (на одно слово).

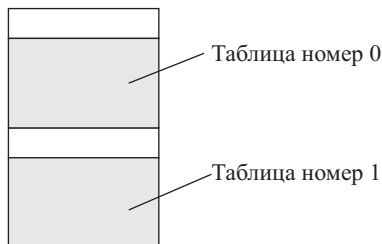
#### **LIFO(634)**

По команде LIFO(634) производится чтение слова, которое было записано в стек последним. Для этого указатель уменьшается на единицу, затем производится чтение данных в указываемом адресе. Прочитанные данные остаются неизменными.



**Команды обработки таблиц записей**

Последовательности данных, состоящие из более чем одной записи и содержащие одинаковое количество слов в каждой из записей, называется таблицей данных. Таблица данных содержится в указанной области памяти ввода/вывода и может регистрироваться в качестве области таблицы с помощью команды DIM. Существует возможность регистрации 16 отдельных таблиц, с номерами от 0 до 15.



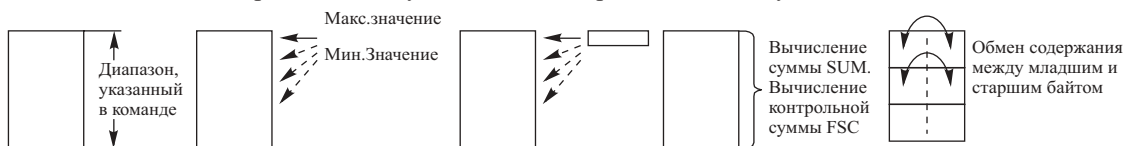
Следующий ниже рисунок показывает структуру таблицы записей. Каждая из записей содержит одинаковое количество слов.



Для косвенного указания таблицы данных может применяться индексный регистр (IR). Вычисление адреса записи легко осуществляется при помощи команды SETR(635) (SET RECORD NUMBER – Задание номера записи) и команды GETR(636) (GET RECORD NUMBER – Получение номера записи).

**Команды выполнения операций в диапазоне данных**

Команды, включенные в данный раздел, выполняют действия в диапазоне слов для нахождения максимального или минимального значения, поиска определенного значения, вычисления контрольной суммы FSC, или обмена содержания между младшим и старшим байтом в указанных словах.

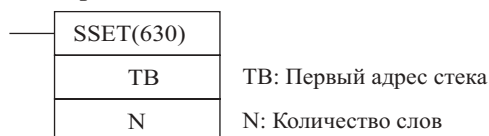




## 3-16-1 Команда задания стека SET STACK: SSET(630)

**Назначение**

По команде SSET(630) определяется стек заданной длины, начиная с указанного слова.

**Символ релейно-контактной схемы****Модификации**

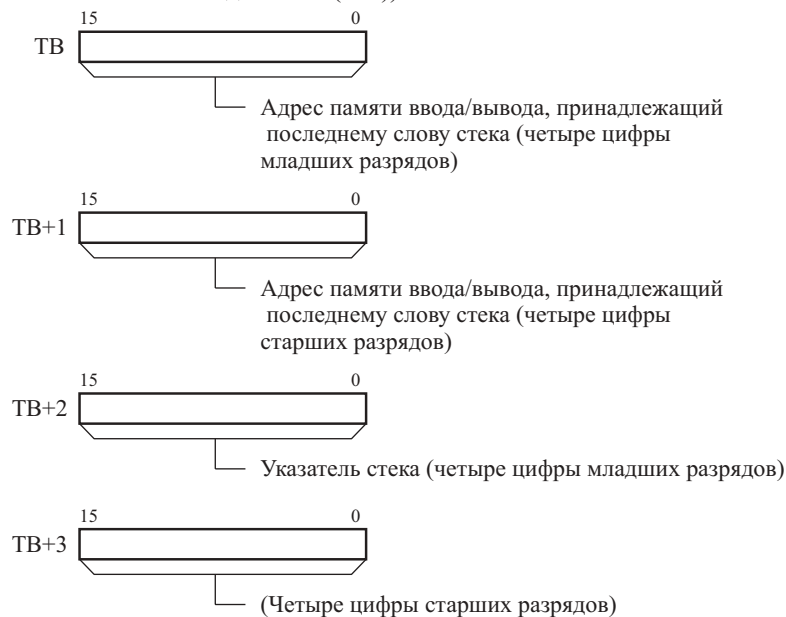
Модификации	Выполняется в каждом цикле для состояния ON	SSET(630)
	Выполняется один раз для дифференцирования вверх	@ SSET(630)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

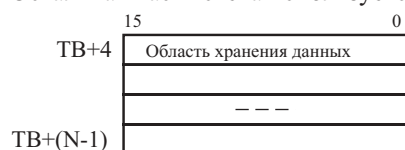
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов****ТВ...ТВ+3: слова управления стеком**

Первые четыре слова стека содержат адрес памяти внутреннего ввода/вывода, принадлежащий последнему слову стека, и указатель стека (адрес следующего слова, которое будет перезаписано при выполнении команды PUSH(632)).

**ТВ+4...ТВ+(N-1): Область хранения данных**

Остальная часть стека используется для хранения данных.



- Примечание:**
1. Начальным значением указателя стека всегда является адрес памяти внутреннего ввода/вывода ТВ+4.
  2. Слова ТВ и ТВ+4 должны находиться в одной области данных.

### Спецификации операндов

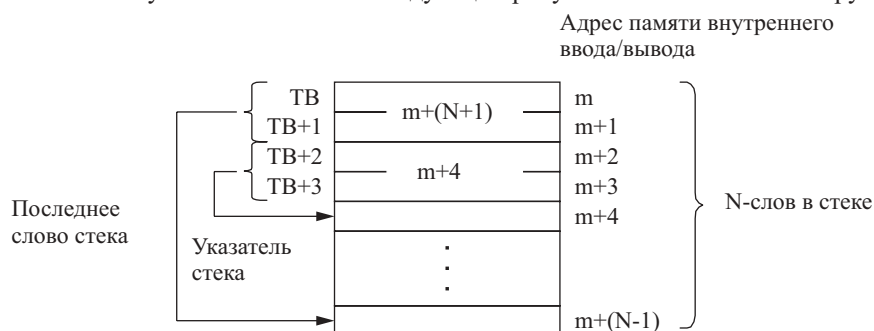
Область	TB	N
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A448...A959	A000...A959
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0... C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)	
Константы	–	#0005 #FFFF (двоичные) или &5...65535
Регистры данных	–	DR0...DR15
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15	

### Описание

По команде SSET(630) создается стек, состоящий из N слов. Стек начинается в TB и заканчивается в TB+(N-1). Первые два слова стека (TB и TB+1) содержат восьмизначный (в шестнадцатеричном коде) адрес памяти ввода/вывода, принадлежащий последнему слову стека.

Следующие два слова (TB+3 и TB+2) содержат указатель стека. Указатель стека- это адрес памяти ввода/вывода, принадлежащий слову, которое будет перезаписано при выполнении команды PUSH(632). Начальное значение указателя стека-это адрес TB+4.

Команда SSET(630) автоматически инициализирует область данных в стеке (от TB+4 до TB+(N-1)), записывая нули в эти слова. На следующем рисунке показана базовая структура стека.



По команде SSET(630) производится только задание и инициализация стека. Для записи и чтения данных в стеке используйте следующие ниже команды.

- 1,2,3...** 1. По команде PUSH(632) производится запись данных в стек, одно слово за один раз.
2. По команде FIFO(633) и LIFO(634) осуществляется чтение данных в стеке. При выполнении команды FIFO(633) читается слово, которое было записано в стек первым, а по команде LIFO(634) чтение слова, которое было занесено в стек последним.
3. Значение указателя стека в слове управления стеком автоматически обновляется при каждом выполнении команд PUSH(632), FIFO(633) и LIFO(634). Обычно пользователь может не беспокоиться о содержании слова управления стеком. При осуществлении доступа к содержанию стека используя другие команды, ус-

манавливайте значение указателя стека косвенным методом, используя индексный регистр (IR).

### Флаги

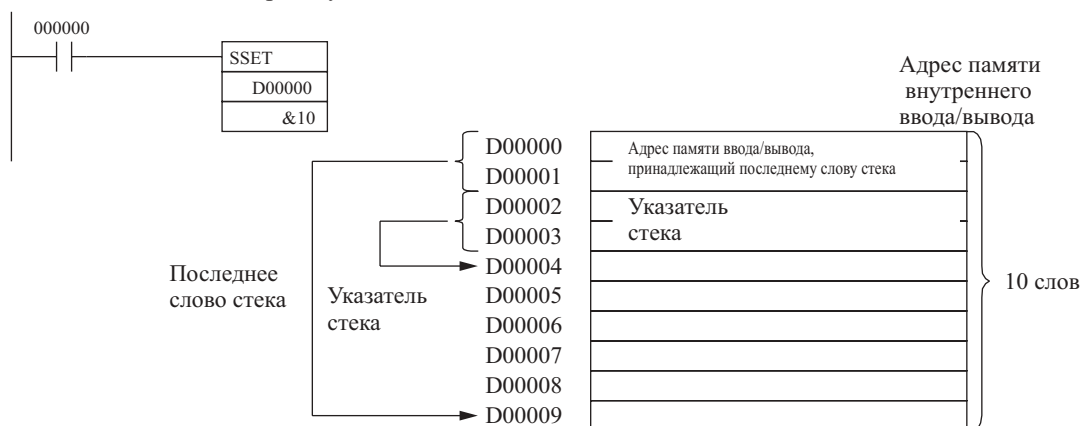
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда N выходит за пределы значений от 0005 до FFFF. OFF в других случаях.

### Меры предосторожности

Минимальная величина для количества слов стека N равна 5, т.к. N включает 4 слова, содержащих указатель последнего слова стека и собственно указатель стека. Если N выходит за пределы от 0005 до FFFF определяется ошибка.

### Примеры

Когда в следующем ниже примере CIO 000000 находится в состоянии ON, по команде SSET(630) создается стек из 10 слов от D00000 до D00009. D00000 и D00001 содержат адрес последнего слова стека. D00002 и D00003 содержат указатель стека. Собственно стек начинается с D00004.

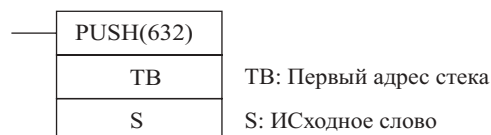


### 3-16-2 Команда записи данных PUSH ONTO STACK: PUSH(632)

#### Назначение

По команде PUSH(632) осуществляется запись одного слова данных в указанный стек.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	PUSH(632)
	Выполняется один раз для дифференцирования вверх	@ PUSH(632)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

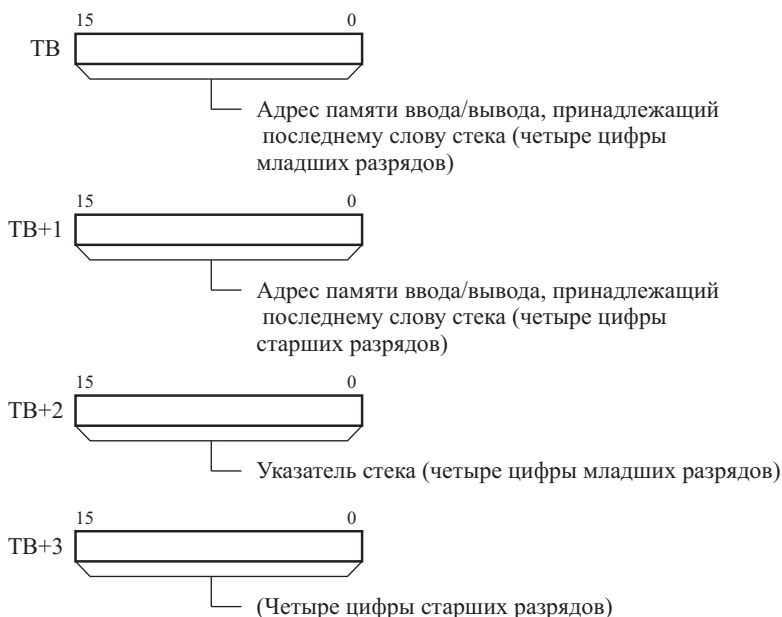
#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

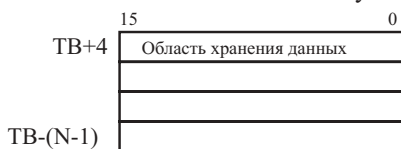
#### Операнды

##### ТВ...ТВ+3: слова управления стеком

Первые четыре слова стека содержат адрес памяти внутреннего ввода/вывода, принадлежащий последнему слову стека, указатель стека (адрес следующего слова, которое будет перезаписано при выполнении команды PUSH(632)).

**ТВ+4...ТВ+(N-1): Область хранения данных**

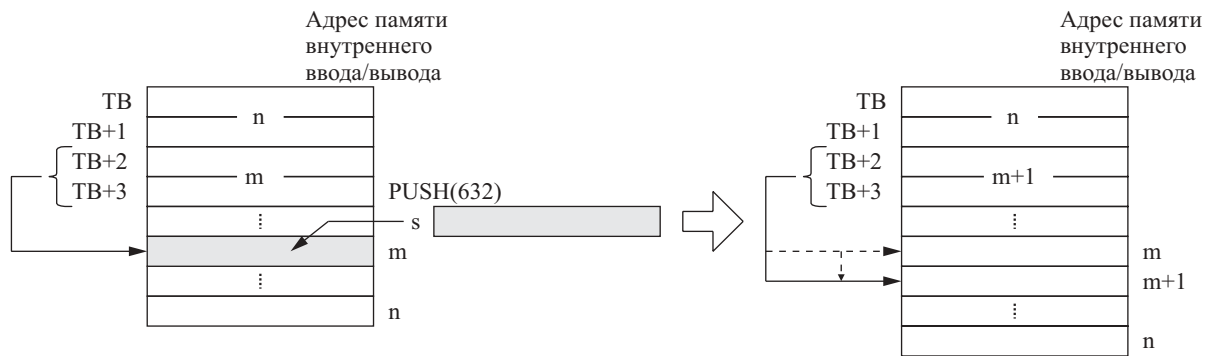
Остальная часть стека используется для хранения данных.

**Спецификации операндов**

Область	ТВ	S
Область ввода/вывода (область СЮ)	СЮ 0000...СЮ 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A448...A959	A000...A959
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0... C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)	
Константы	—	#0000 #FFFF (двоичные) или &5...65535
Регистры данных	—	DR0...DR15
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15	

**Описание**

По команде PUSH(632) осуществляется запись содержание слова S в адрес определяемый указателем стека (ТВ+3, ТВ+2), и увеличивает указатель стека на единицу.



После использования команды PUSH(632) и записи данных в стек чтение данных в стеке можно производить с помощью команд FIFO(632) и LIFO(633).

**Флаги**

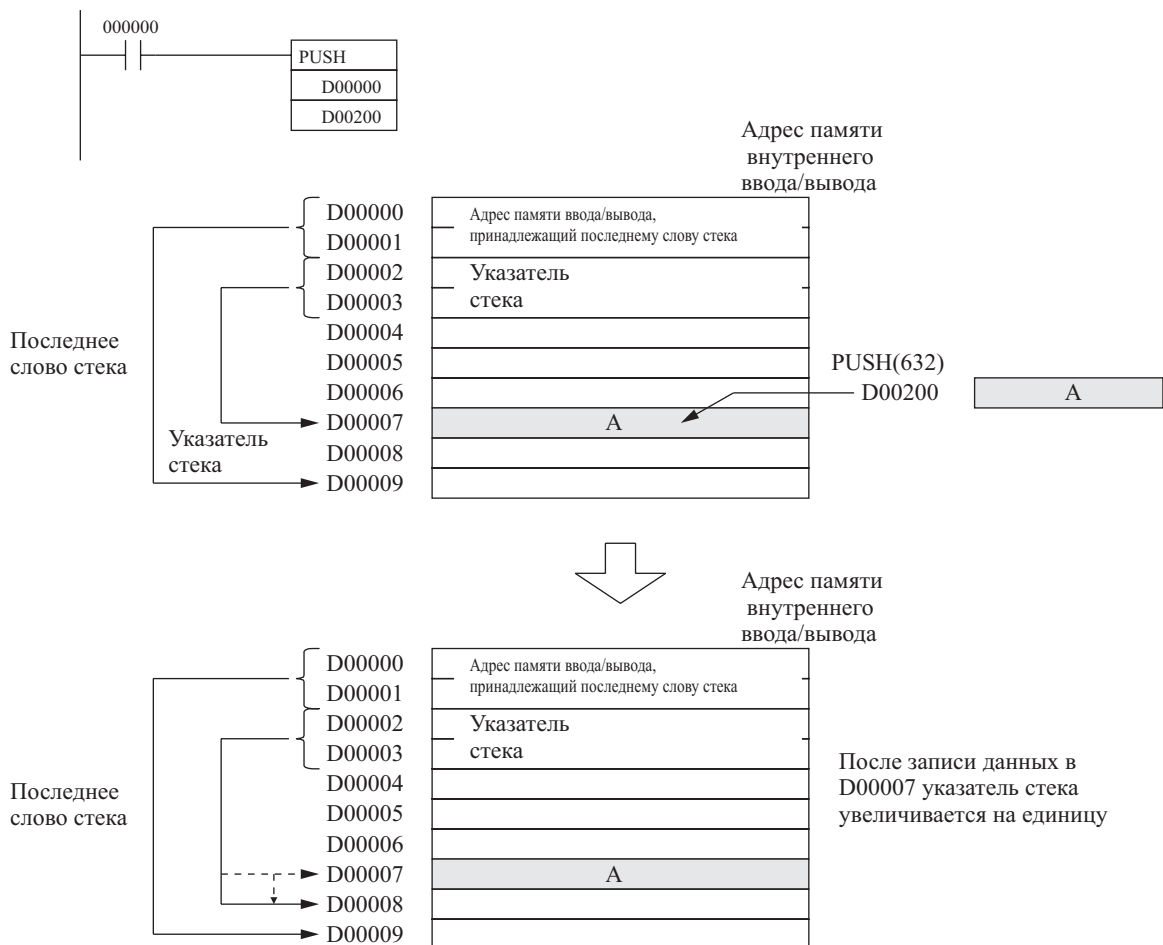
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда адрес, определяемый указателем стека (TB+3 и TB+2), превышает адрес последнего слова стека. OFF в других случаях.

**Меры предосторожности**

Стек должен быть предварительно определен при помощи команды SSET(630)

**Примеры**

Когда в следующем ниже примере CIO 000000 находится в состоянии ON, по команде PUSH(632) производится запись содержания D00200 в стек, начиная с D00000. В данном случае указатель стека указывает на D00007.



## 3-16-3 Команда чтения первой записи FIRST IN FIRST OUT: FIFO(633)

**Назначение**

По команде FIFO(633) осуществляется чтение одного слова данных, записанного в указанный стек первым (наиболее старые данные стека).

**Символ релейно-контактной схемы**

—	FIFO(633)	
	ТВ	ТВ: Первый адрес стека
	D	D: Слово назначения

**Модификации**

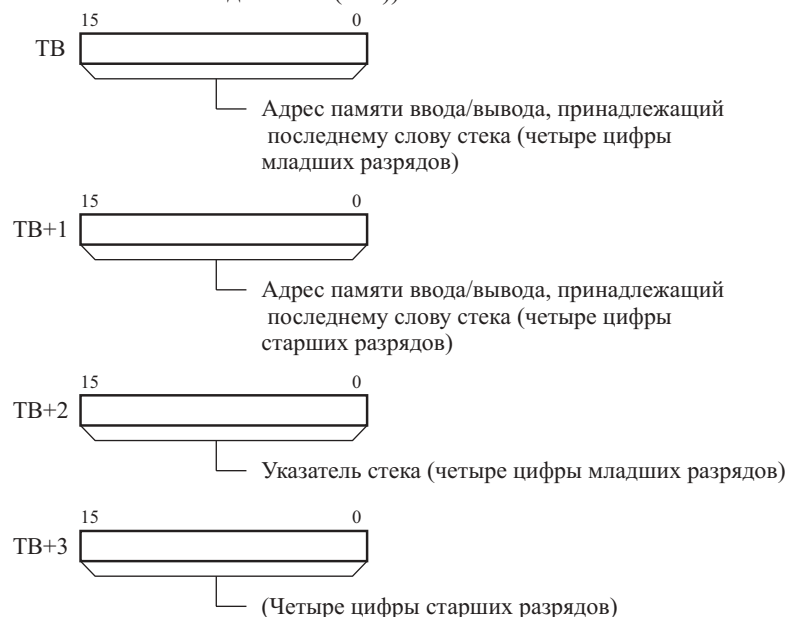
Модификации	Выполняется в каждом цикле для состояния ON	FIFO(633)
	Выполняется один раз для дифференцирования вверх	@ FIFO(633)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

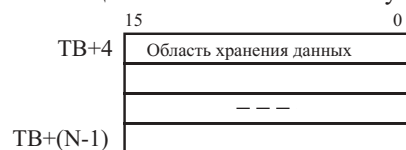
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****ТВ...ТВ+3: слова управления стеком**

Первые четыре слова стека содержат адрес памяти внутреннего ввода/вывода, принадлежащий последнему слову стека, указатель стека (адрес следующего слова, которое будет перезаписано при выполнении команды PUSH(632)).

**ТВ+4...ТВ+(N-1): Область хранения данных**

Остающаяся часть стека используется для хранения данных.

**Спецификации операндов**

Область	ТВ	D
Область ввода/вывода (область СЮ)	СЮ 0000...СЮ 6143	

Область	TB	D
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A448...A959	
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0... C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)	
Константы	—	
Регистры данных	—	DR0...DR15
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-)IR15	

### Описание

По команде FIFO(633) осуществляется чтение наиболее старого слова данных в стеке (TB+4) и запись этих данных в D. После этого указатель стека (TB+3, TB+2) и уменьшается на единицу, а все оставшиеся данные перемещаются вниз на одно слово. После прочтения данных, содержание TB+4 удаляется. Данные в конце стека (адрес, который был указан указателем стека) остаются неизменными.



Используйте команду FIFO(633) в комбинации с командой PUSH(632). После использования команды PUSH(632) и записи данных в стек, чтение данных в стеке можно производить с помощью команды FIFO(632) по принципу «Первыми введены, первыми прочитаны».

По команде FIFO(633) производится чтение начальных данных стека и удаление этих данных с одновременным передвижением вперед следующих данных.

### Флаги

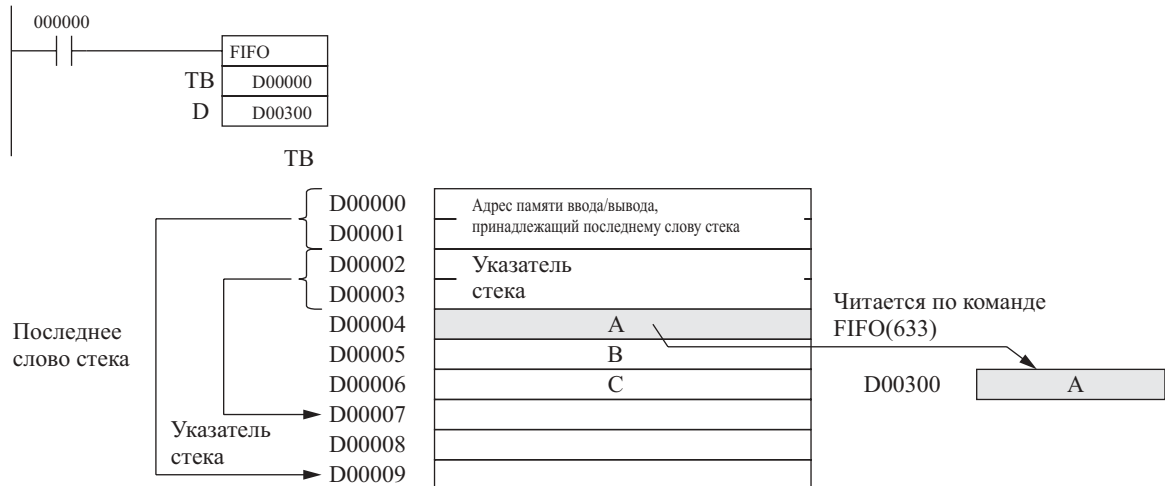
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда адрес, определяемый указателем стека (TB+3 и TB+2), меньше или равен адресу первого слова в области хранения данных в стеке (TB+4). (Такая ситуация является ошибкой отрицательного переполнения стека.) OFF в других случаях.

**Меры предосторожности**

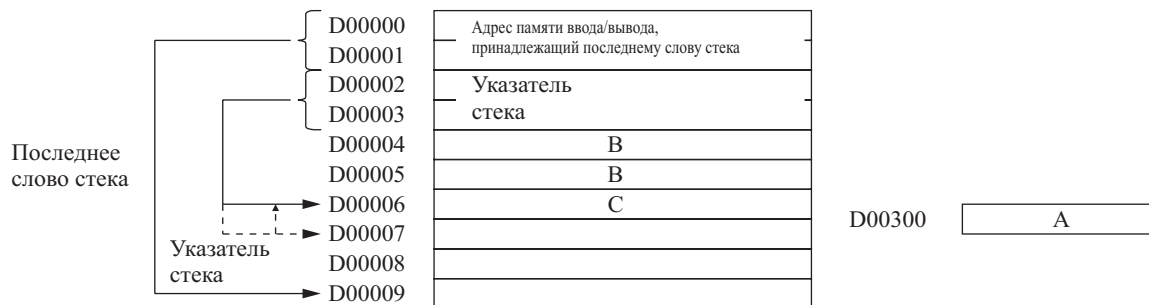
Стек должен быть предварительно определен при помощи команды SSET(630)

**Примеры**

Когда в следующем ниже примере CIO 000000 находится в состоянии ON, по команде FIFO(633) производится чтение содержания D00004 (ТВ+4 для стека начинающегося с D00000). Данные записываются в D00300.



После записи данных в D00300 указатель стека уменьшается на единицу, а оставшиеся данные смещаются вниз. (Содержание D00005 перемещается в D00004, а содержание D00006 перемещается в D00005.)

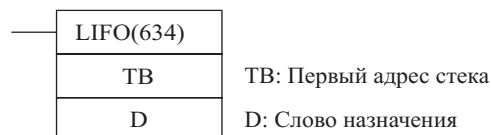


**3-16-4 Команда чтения последней записи LAST IN FIRST OUT: LIFO(634)**

**Назначение**

По команде LIFO(634) осуществляется чтение одного слова или данных, записанных в указанный стек позднее других (наиболее свежие данные стека).

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	LIFO(634)
	Выполняется один раз для дифференцирования вверх	@ LIFO(634)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

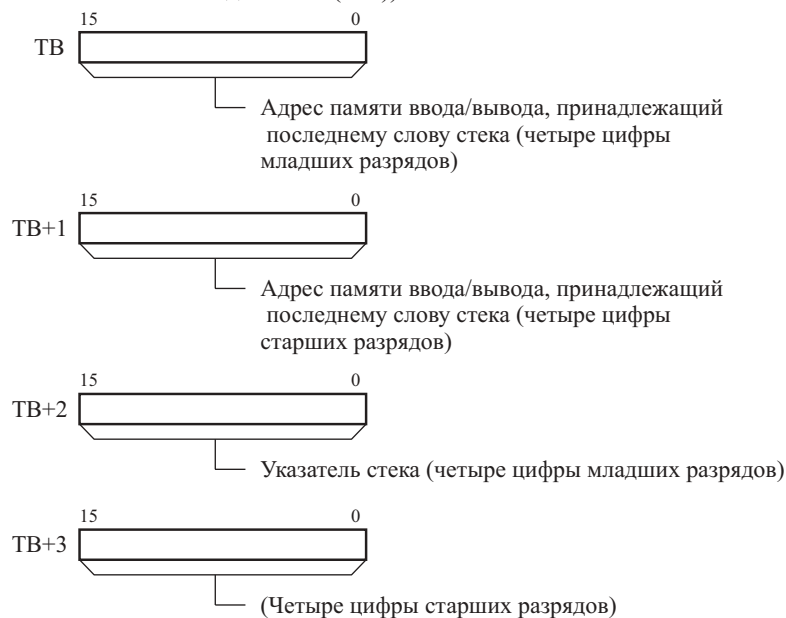


**Применяемые области программы**

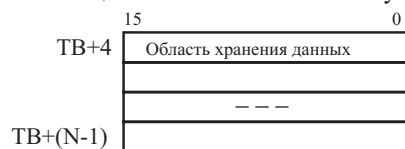
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****ТВ...ТВ+3: слова управления стеком**

Первые четыре слова стека содержат адрес памяти внутреннего ввода/вывода, принадлежащий последнему слову стека, указатель стека (адрес следующего слова, которое будет перезаписано при выполнении команды PUSH(632)).

**ТВ+4...ТВ+(N-1): Область хранения данных**

Остающаяся часть стека используется для хранения данных.

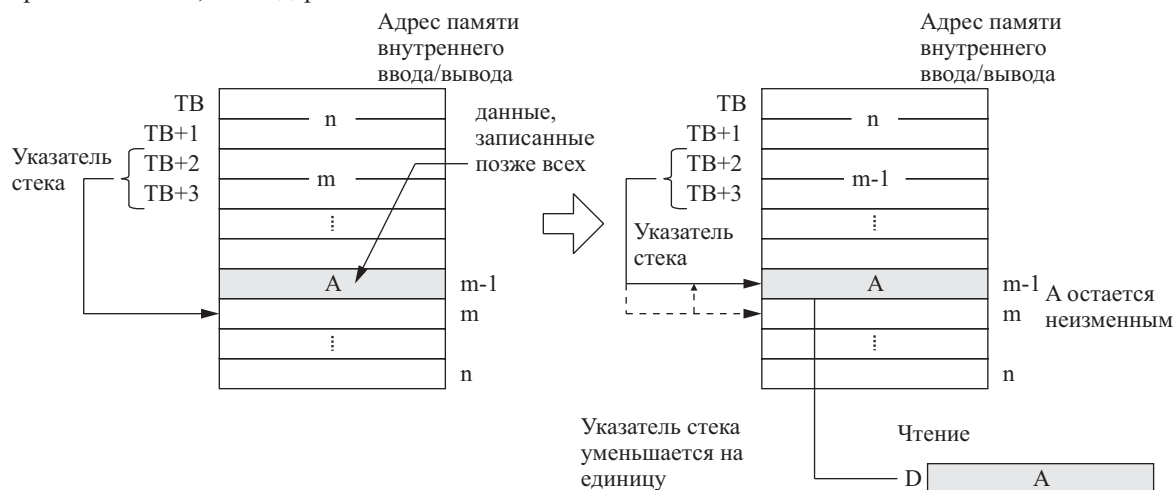
**Спецификации операндов**

Область	ТВ	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A448...A959	
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0... C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)	
Константы	—	
Регистры данных	—	DR0...DR15
Индексные регистры	—	

Область	TB	D
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15	

### Описание

По команде LIFO(634) осуществляется чтение данных из адреса, определяемого указателем стека (слово, записанное последним), уменьшение указателя на единицу и запись прочитанных данных в D. После прочтения слова, его содержание остается в стеке неизменным.



Используйте команду LIFO(634) в комбинации с командой PUSH(632). После использования команды PUSH(632) и записи данных в стек, чтение данных в стеке можно производить с помощью команд LIFO(634) по принципу «Последними введены, первыми прочитаны».

### Флаги

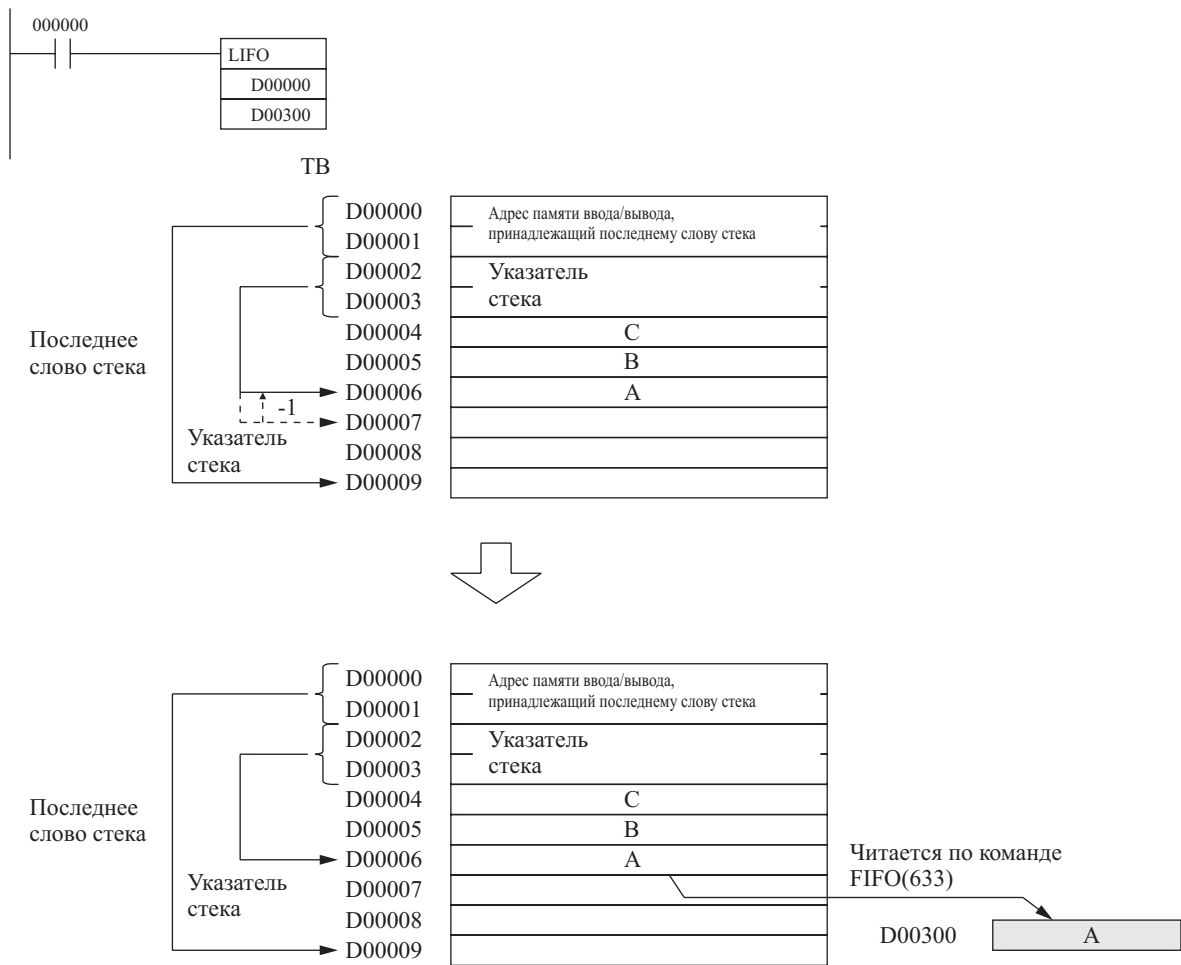
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда адрес, определяемый указателем стека (TB+3 и TB+2), меньше или равен адресу первого слова в области хранения данных в стеке (TB+4). (Такая ситуация является ошибкой отрицательного переполнения стека.) OFF в других случаях.

### Меры предосторожности

Стек должен быть предварительно определен при помощи команды SSET(630)

### Примеры

Когда в следующем ниже примере СЮ 000000 находится в состоянии ON, по команде LIFO(634) производится чтение содержания слова, отмеченного указателем стека (D00006). Данные записываются в D00300.



После записи данных в D00300 указатель стека уменьшается на единицу. Содержание D00006 остается неизменным.

### 3-16-5 Команда задания таблицы записей DIMENSION RECORD TABLE: DIM(631)

#### Назначение

По команде DIM(631) указанные области памяти ввода/вывода определяются в качестве таблицы записей. При этом задается длина записей и количество записей. Одновременно может создаваться до 16-ти таблиц записей.

#### Символ релейно-контактной схемы

DIM(631)	
N	N: Номер таблицы
LR	LR: Длина каждой из записей
NR	NR: Количество записей
TB	TB: Первое слово таблицы

#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	DIM(631)
	Выполняется один раз для дифференцирования вверх	@ DIM(631)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****N: Номер таблицы**

Определяет номер таблицы. N должен находиться в пределах от 0 до 15.

**LR: Длина каждой из записей**

Обозначает количество слов в каждой из записей. Значение LR должно находиться в пределах от 0001 до FFFF (шестн.) (от 1 до 65535 в десятичном коде).

**NR: Количество записей**

Указывает количество записей в таблице. Значение NR должно находиться в пределах от 0001 до FFFF (шестн.) (от 1 до 65535 в десятичном коде).

**TB: Первое слово таблицы**

Указывает первое слово таблицы. Все слова в таблице должны находиться в одной области данных. Другими словами TB, TB+LR...NR-1 должны находиться в одной области данных.

**Спецификации операндов**

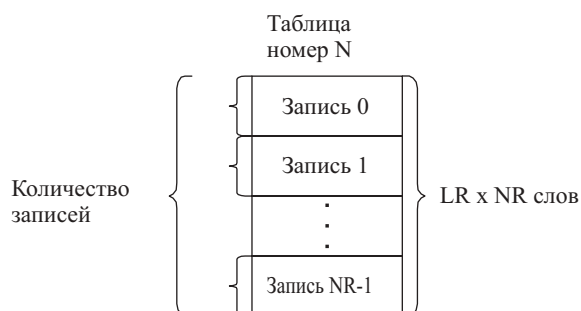
Область	N	LR	NR	D
Область ввода/вывода (область CIO)	–	CIO 0000...CIO 6143		
Рабочая область	–	W000...W511		
Область удержания бита (Holding bit area)	–	H000...H511		
Область вспомогательных битов	–	A000...A959	A448...A959	
Область таймера	–	T0000...T4095		
Область счетчика	–	C0000...C4095		
Область DM	–	D00000...D32767		
Область EM, не содержащая банков	–	E00000...E32767		
Область EM, содержащая банки	–	En_00000...En_32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	–	@D00000...@D32767 @E00000...@E32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)		
Константы	0...15	#0001...#FFFF (двоичные) или &1...&65535	–	
Регистры данных	–	DR0...DR15	–	
Индексные регистры	–	–	–	
Косвенная адресация с использованием индексных регистров	–	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15		

**Описание**

По команде DIM(631) осуществляется регистрация слов от TB до TB+LR...NR-1 в качестве таблицы, имеющей номер N. Таблица номер N содержит NR записей, длина которых равна LR. После того, как область регистрируется в качестве таблицы записей, изменения данных в этой области производиться не могут.

Для упрощения вычисления адресов в таблице данных применяйте команду DIM(631) в комбинации с командами SETR(635) (SET RECORD NUMBER) или GETR(636) (GET RECORD NUMBER). Для разделения данных на записи используйте команду DIM(631), а затем с помощью команды SETR(635) записывайте первый адрес желаемой записи в индексный регистр. После этого индексный регистр может использоваться в качестве указателя в других командах, например при чтении, записи, поиске или сравнении данных.

Например, если значения температуры, давления или другой набор параметров сохраняется в виде записей, а записи для различных моделей комбинируются в таблицу, таким образом, облегчается чтение набора параметров для каждой из моделей и для любых отдельных условий.



С командой DIM(631) связаны две команды для таблиц записей: SETR(635) и GETR(636). Команда SETR(635) устанавливает начальный адрес указанной записи в заданный индексный регистр. Команда GETR(636) выводит номер записи индексного регистра, которая содержит заданный адрес (адрес памяти внутреннего ввода/вывода).

#### Флаги

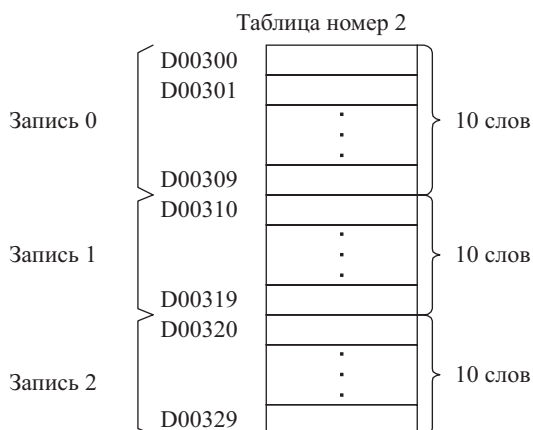
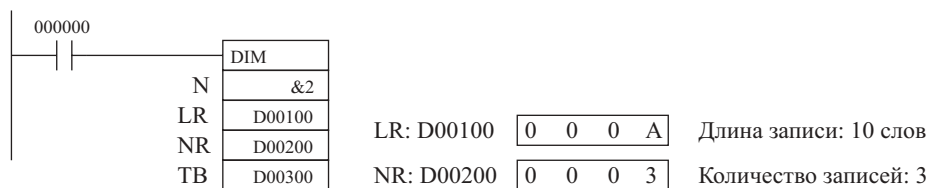
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение LR или NR равно 0000. OFF в других случаях.

#### Меры предосторожности

Записи в зарегистрированной таблице идентифицируются согласно их номерам, диапазон значений от 0 до NR-1.

#### Примеры

Когда в следующем ниже примере CIO 000000 находится в состоянии ON, команда DIM(0631) определяет таблицу записей номер 2, содержащую три записи по 10 слов. Таблица начинается с D00300.



### 3-16-6 Команда указания расположения записи SET RECORD LOCATION: SETR(635)

#### Назначение

По команде SETR(635) производится запись в указанный индексный регистр места расположения определенной записи (т.е. адрес памяти внутреннего обмена, соответствующий началу указанной записи).

**Символ релейно-контактной схемы**

—	SETR(635)	
	N	N: Номер таблицы
	R	R: Номер записи
	D	D: Индексный регистр назначения

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	SETR(635)
	Выполняется один раз для дифференцирования вверх	@ SETR(635)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****N: Номер таблицы**

Определяет номер таблицы. N должен находиться в пределах от 0 до 15.

**R: Номер записи**

Обозначает номер желаемой записи. Значение R должно находиться в пределах от 0001 до FFFF (шестн.) (от 1 до 65535 в десятичном коде). Номера записей начинаются с 0, следовательно, для таблицы из NR записей действительными номерами являются 0...NR-1.

**D: индексный регистр назначения**

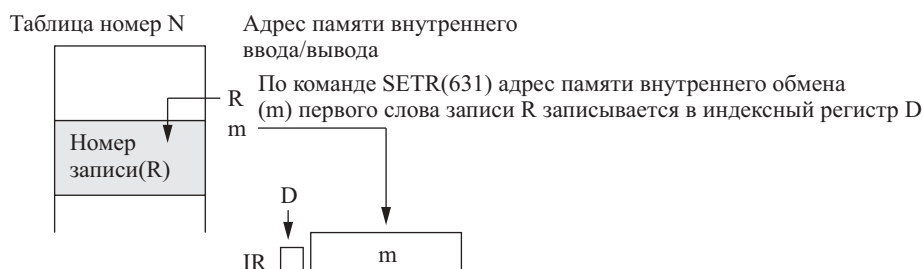
Указывает желаемый индексный регистр. Значения D должны находиться в пределах от IR0 до IR15.

**Спецификации операндов**

Область	N	R	D
Область ввода/вывода (область CIO)	—	CIO 0000...CIO 6143	—
Рабочая область	—	W000...W511	—
Область удержания бита (Holding bit area)	—	H000...H511	—
Область вспомогательных битов	—	A000...A959	—
Область таймера	—	T0000...T4095	—
Область счетчика	—	C0000...C4095	—
Область DM	—	D00000...D32767	—
Область EM, не содержащая банков	—	E00000...E32767	—
Область EM, содержащая банки	—	En_00000...En_32767 (n = 0... C)	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767	—
Константы	0...15	#0001...#FFFF(двоичные) &1...&65535	или —
Регистры данных	—	DR0...DR15	—
Индексные регистры	—	—	IR0...IR15
Косвенная адресация с использованием индексных регистров	—	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(--)IR0...,-(--)IR15	—

**Описание**

По команде SETR(635) производится запись адреса памяти внутреннего обмена, соответствующего началу определенной записи, в указанный индексный регистр. Следующий ниже рисунок показывает работу команды SETR(631).

**Флаги**

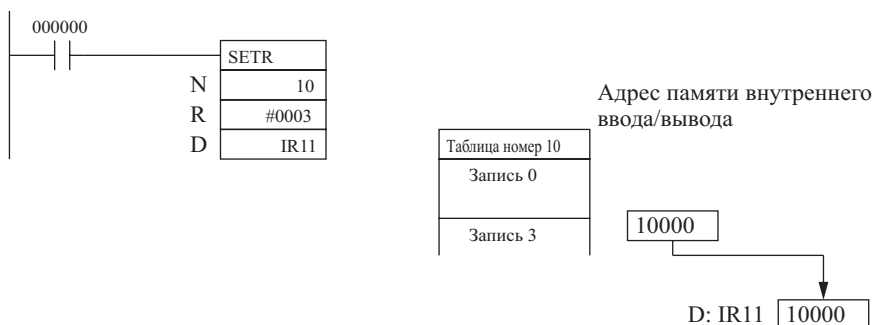
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда номер таблицы предварительно не определен командой DIM(631). Переводится в состояние ON, когда указываемый номер записи R превышает максимальный номер записи в таблице NR-1. OFF в других случаях.

**Меры предосторожности**

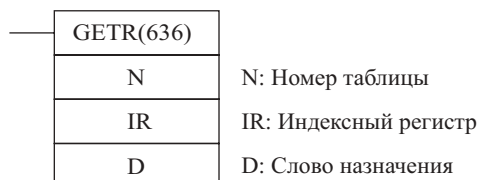
Таблица должна быть предварительно определена командой DIM(631). Номера записей должны находиться в пределах от 0 до NR-1, где NR – это количество записей, указанных при определении таблицы по команде DIM(631).

**Примеры**

Когда в следующем ниже примере СЮ 000000 находится в состоянии ON, команда SETR(635) находит адрес первого слова записи 3 в таблице номер 10 и записывает этот адрес в индексный регистр IR11.

**3-16-7 Команда чтения расположения записи GET RECORD NUMBER: GETR(636)****Назначение**

По команде GETR(636) в указанном индексном регистре производится нахождение номера записи, содержащей адрес памяти внутреннего ввода/вывода.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	GETR(636)
	Выполняется один раз для дифференцирования вверх	@ GETR(636)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****N: Номер таблицы**

Обозначает номер таблицы. N должен находиться в пределах от 0 до 15.

**IR: Индексный регистр**

Указывает желаемый индексный регистр. Значения D должны находиться в пределах от IR0 до IR15.

**D: Слово назначения**

Указывает слово, в которое будет записан номер записи.

**Спецификации операндов**

Область	N	IR	D
Область ввода/вывода (область CIO)	–		CIO 0000...CIO 6143
Рабочая область	–		W000...W511
Область удержания бита (Holding bit area)	–		H000...H511
Область вспомогательных битов	–		A448...A959
Область таймера	–		T0000...T4095
Область счетчика	–		C0000...C4095
Область DM	–		D00000...D32767
Область EM, не содержащая банков	–		E00000...E32767
Область EM, содержащая банки	–		En_00000...En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	–		@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–	)	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)
Константы	0...15	–	–
Регистры данных	–		DR0...DR15
Индексные регистры	–	IR0...IR15	–
Косвенная адресация с использованием индексных регистров	–		,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++), ,IR15(++) ,(- -)IR0 ,(- -)IR15

**Описание**

По команде GETR(636) в указанном индексном регистре определяется запись, которая содержит адрес памяти внутреннего обмена. Номер этой записи заносится в слово D. Адрес, содержащийся в индексном регистре, не обязательно является первым словом записи, он может принадлежать любому слову записи. Следующий ниже рисунок поясняет работу команды GETR(636).





**Флаги**

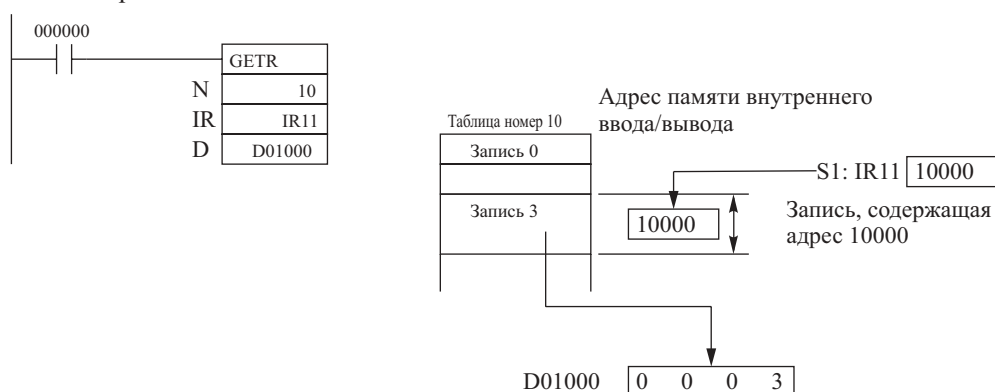
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда адрес памяти внутреннего обмена, содержащийся в индексном регистре, находится за пределами указанной таблицы. Переводится в состояние ON, когда номер указываемой таблицы не определен командой DIM(631). OFF в других случаях.

**Меры предосторожности**

Таблица должна быть предварительно определена командой DIM(631), а адрес памяти внутреннего ввода/вывода, содержащийся в индексном регистре, должен находиться в пределах указанной таблицы.

**Примеры**

Когда в следующем ниже примере СЮ 000000 находится в состоянии ON, команда GETR(636) находит номер записи, которая содержит адрес памяти внутреннего обмена в индексном регистре IR11. Найденный номер записывается в D01000.

**3-16-8 Команда поиска данных DATA SEARCH: SRCH(181)****Назначение**

По команде SRCH(181) производится поиск одного слова данных в диапазоне слов.

**Символ релейно-контактной схемы**

SRCH(181)	
C	C: Первое управляющее слово
R1	R1: Первое слово в диапазоне слов
Cd	Cd: Данные сравнения

**Модификации**

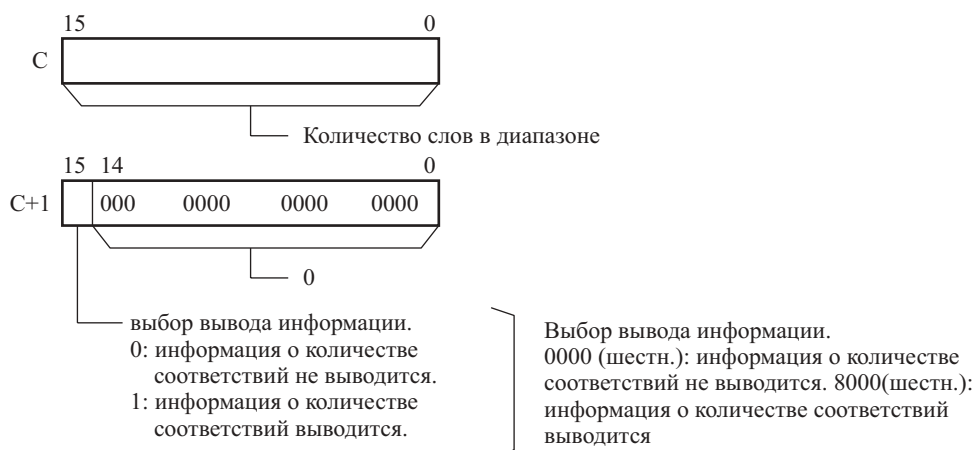
Модификации	Выполняется в каждом цикле для состояния ON	SRCH(181)
	Выполняется один раз для дифференцирования вверх	@ SRCH(181)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****С и С+1: Управляющие слова**

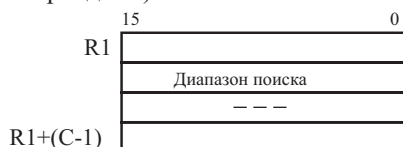
Слово С указывает количество слов в диапазоне, а бит 15 слова С+1 указывает выводить или не выводить количество соответствий в DR00.



**Примечание:** Слова  $C$  и  $C+1$  должны находиться в одной области данных.

### R1: Первое слово в диапазоне слов

R1 определяет первое слово диапазона слов, где осуществляется поиск. Слова от R1 до R1+(C-1) просматриваются на предмет поиска желаемых данных. (C- это количество слов, задаваемых в операнде C.)



**Примечание:** Слова  $R1 \dots R1+(C-1)$  должны находиться в одной области данных.

### Спецификации операндов

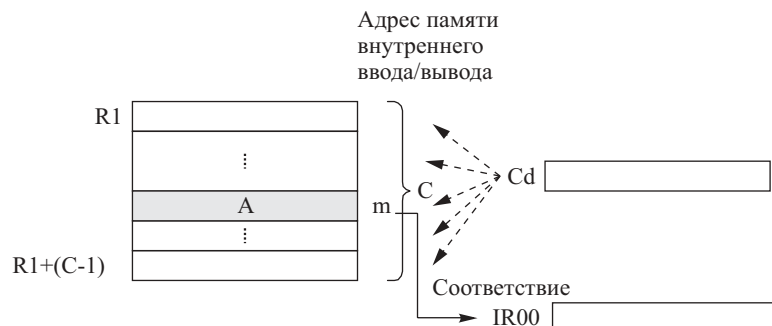
Область	C	R1	Cd
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	CIO 0000...CIO 6143	
Рабочая область	W000...W510	W000...W511	
Область удержания бита (Holding bit area)	H000...H510	H000...H511	
Область вспомогательных битов	A000...A958	A448...A959	
Область таймера	T0000...T4094	T0000...T4095	
Область счетчика	C0000...C4094	C0000...C4095	
Область DM	D00000...D32766	D00000...D32767	
Область EM, не содержащая банков	E00000...E32766	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32766 (n = 0... C)	En_00000...En_32767 (n = 0... C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)		
Константы	Только указанные значения	—	#0000...#FFFF (двоичные)
Регистры данных		DR0...DR15	
Индексные регистры			
Косвенная адресация с использованием индексных регистров		,IR0...,IR15 -2048...+2047,IR0 -2048...+2047, IR15 DR0...DR15, IR0...IR15 ,IR0+(++) ,IR15+(++) ,(- -)IR0 ,(- -)IR15	

### Описание

По команде SRCH(181) осуществляется просмотр области памяти от R1 до R1+(C-1) и поиск в этом диапазоне слов, содержащих данные сравнения. Когда определяется слово, содержащее данные сравнения, ад-

рес этого слова записывается в IR00, а флаг равенства переводится в состояние ON. (Если определяется два и более совпадений, в индексный регистр записывается адрес первого слова, содержащего данные сравнения.)

Если бит 15 слова C+1 установлен в состояние 1, при выполнении команды SRCH(181), кроме того, в регистр данных DR00 записывается количество совпадений. Если бит 15 слова C+1 равен нулю, содержание DR00 не изменяется.



Команда SRCH(181) осуществляет просмотр табличных данных, которые содержат одно слово в каждой из записей. Для просмотра данных, состоящих более чем из одного слова, используйте с индексным регистром (IR) команды DIM(631), SETR(635), GETR(636), FOR(512)-NEXT(513) или BREAK(514).

Для определения наличия искомых данных состояние флага равенства может проверяться немедленно после выполнения команды.

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание C, находится за пределами значений от 0001 до FFFF. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда одно или более слов в указанном диапазоне слов содержат данные сравнения. OFF в других случаях.

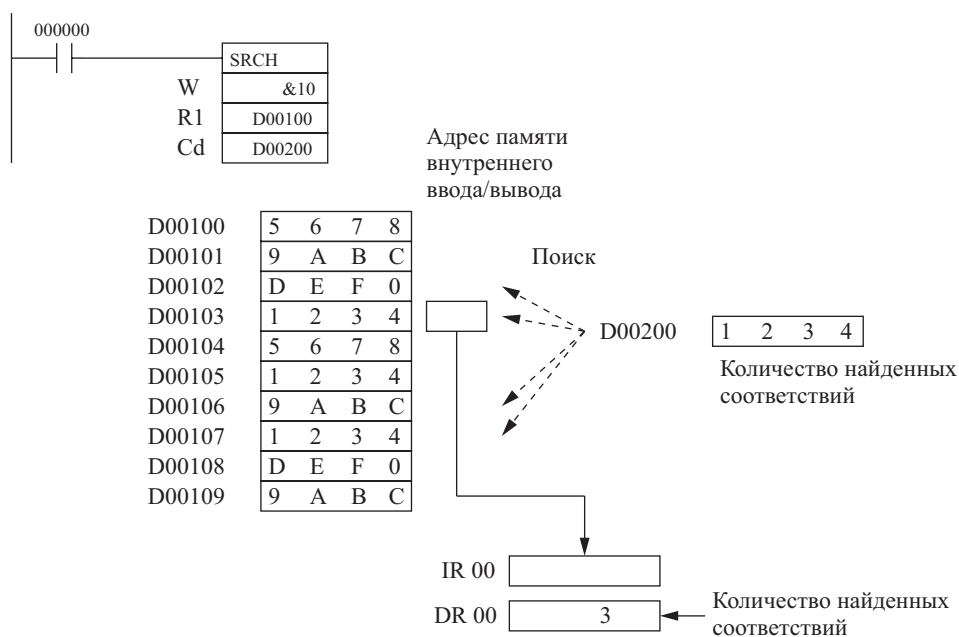
#### Меры предосторожности

В случае, когда совпадение не находится, содержание регистров IR00 и DR00 не изменяется.

#### Примеры

Когда в следующем ниже примере CIO 000000 находится в состоянии ON, по команде SRCH(181) производится просмотр 10 слов, начиная с D00100, с целью поиска слов, содержание которых аналогично содержанию D00200.

Адрес первого слова, содержащего данные сравнения, записывается в IR00, а общее количество найденных соответствий записывается в DR00.

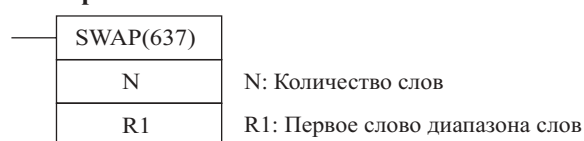


### 3-16-9 Команда перестановки байтов SWAP BYTES: SWAP(637)

#### Назначение

По команде SWAP(637) осуществляется перестановка местами младшего и старшего байтов во всех словах в указанном диапазоне слов.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	SWAP(637)
	Выполняется один раз для дифференцирования вверх	@ SWAP(637)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

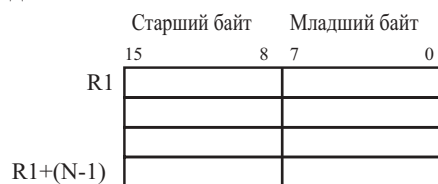
#### Операнды

##### N: Количество слов

N указывает количество слов в заданном диапазоне и должно находиться в пределах от 0001 до FFFF (шестн.) (т.е. от &1 до & 65535).

##### R1: Первое слово диапазона слов

R указывает первое слово заданного диапазона. R1 и R+(N-1) должны находиться в одной области данных.



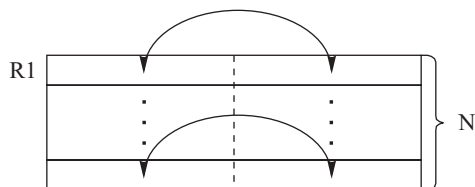
### Спецификации операндов

Область	N	R1
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A448...A959	A000...A959
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n = 0... C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0... C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)	
Константы	#0000 #FFFF (двоичные) или &5...65535	—
Регистры данных	DR0...DR15	—
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,-(-)IR0...,(-)IR15	

### Описание

По команде SWAP(637) производится изменение места расположения между двумя байтами всех слов в диапазоне от R1 до R1+N-1. Эта команда может применяться для изменения последовательности расположения в словах символов ASCII кода.

Расположение байтов изменяется

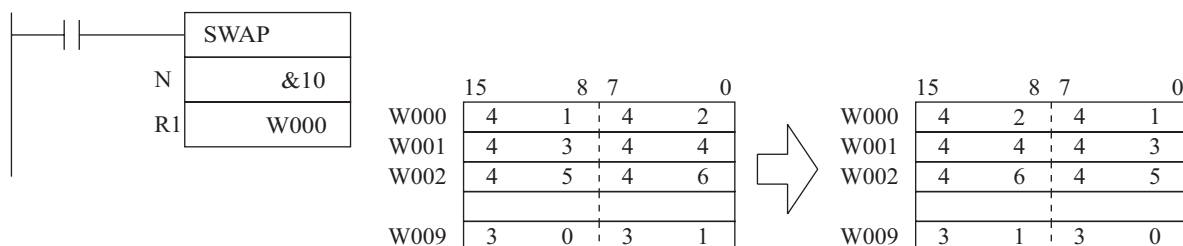


### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда N равен 0000. OFF в других случаях.

### Примеры

Когда в следующем ниже примере CIO 000000 находится в состоянии ON, по команде SWAP(637) производится изменение расположения между байтами в каждом из слов, в диапазоне из 10 слов от W000 до W009.



## 3-16-10 Команда нахождения максимума FIND MAXIMUM: MAX(182)

**Назначение**

По команде MAX(182) производится поиск максимального значения в заданном диапазоне слов.

**Символ релейно-контактной схемы**

MAX(182)	
C	C: Первое управляющее слово
R1	R1: Первое слово в диапазоне слов
D	D: Слово назначения

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	MAX(182)
	Выполняется один раз для дифференцирования вверх	@ MAX(182)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

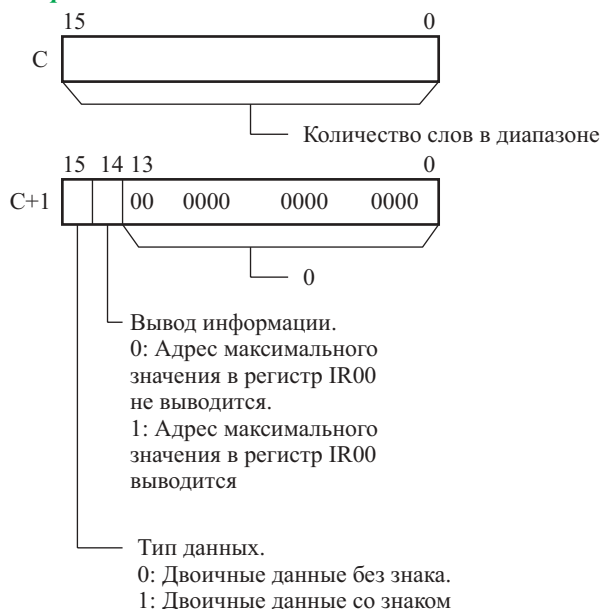
**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****С и С+1: Управляющие слова**

С указывает количество слов в диапазоне. Бит 15 слова С+1 указывает, в каком виде производится обработка данных: в виде двоичных данных со знаком или в виде двоичных данных без знака. Бит 14 слова С+1 указывает, будет ли выводиться в индексный регистр IR00 адрес слова, содержащего максимальное значение.

**Примечание:** Слова С и С+1 должны находиться в одной области данных.

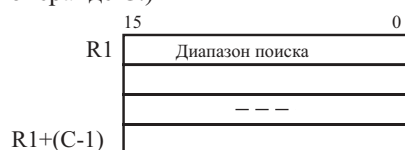


В следующей ниже таблице показаны возможные значения С,

С+1	Тип данных	Запись адреса
0000	Двоичные данные без знака	Нет
4000	Двоичные данные без знака	Да
8000	Двоичные данные со знаком	Нет
C000	Двоичные данные со знаком	Да

**R1: Первое слово в диапазоне слов**

R1 Указывает первое слово диапазона слов, где осуществляется поиск. Слова от R1 до R1+(C-1) просматриваются на предмет поиска максимального значения. (C- это количество слов, задаваемых в операнде C.)



**Примечание:** Слова R1...R1+(C-1) должны находиться в одной области данных.

**Спецификации операндов**

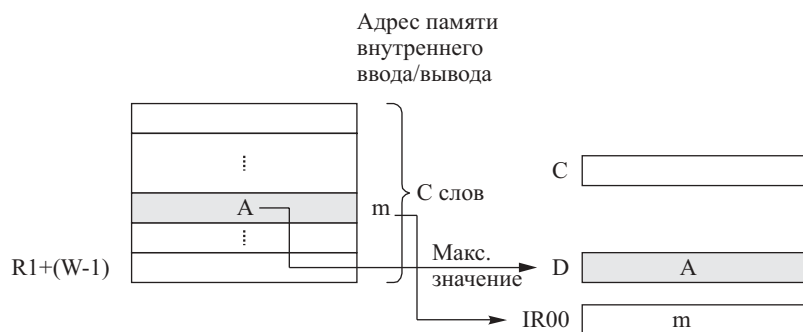
Область	C	R1	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	CIO 0000...CIO 6143	
Рабочая область	W000...W510	W000...W511	
Область удержания бита (Holding bit area)	H000...H510	H000...H511	
Область вспомогательных битов	A000...A958	A000...A959	A448...A959
Область таймера	T0000...T4094	T0000...T4095	
Область счетчика	C0000...C4094	C0000...C4095	
Область DM	D00000...D32766	D00000...D32767	
Область EM, не содержащая банков	E00000...E32766	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32766 (n = 0... C)	En_00000...En_32767 (n = 0... C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)		
Константы	Только указанные значения	—	
Регистры данных	—	DR0...DR15	
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++) ,IR15+(++) ,(- -)IR0 ,(- -)IR15		

**Описание**

По команде MAX(182) осуществляется просмотр области памяти от R1 до R1+(C-1) и поиск в этом диапазоне слова, содержащего максимальное значение. После нахождения максимального значения оно записывается в D.

Если бит 14 слова C+1 установлен в состояние 1, при выполнении команды MAX(182) адрес слова, содержащего максимальное значение, записывается в индексный регистр IR00. (Если в заданном диапазоне содержится два и более максимальных значений, в индексный регистр записывается адрес первого из слов, содержащих максимальное значение.)

Если бит 15 слова C+1 равен 1, команда MAX(182) обращается с данными в заданном диапазоне как с двоичными данными со знаком.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание C, находится за пределами значений от 0001 до FFFF. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, если максимальное значение равно 0000. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, если бит 15 слова, содержащего максимальное значение, равен 1. OFF в других случаях.

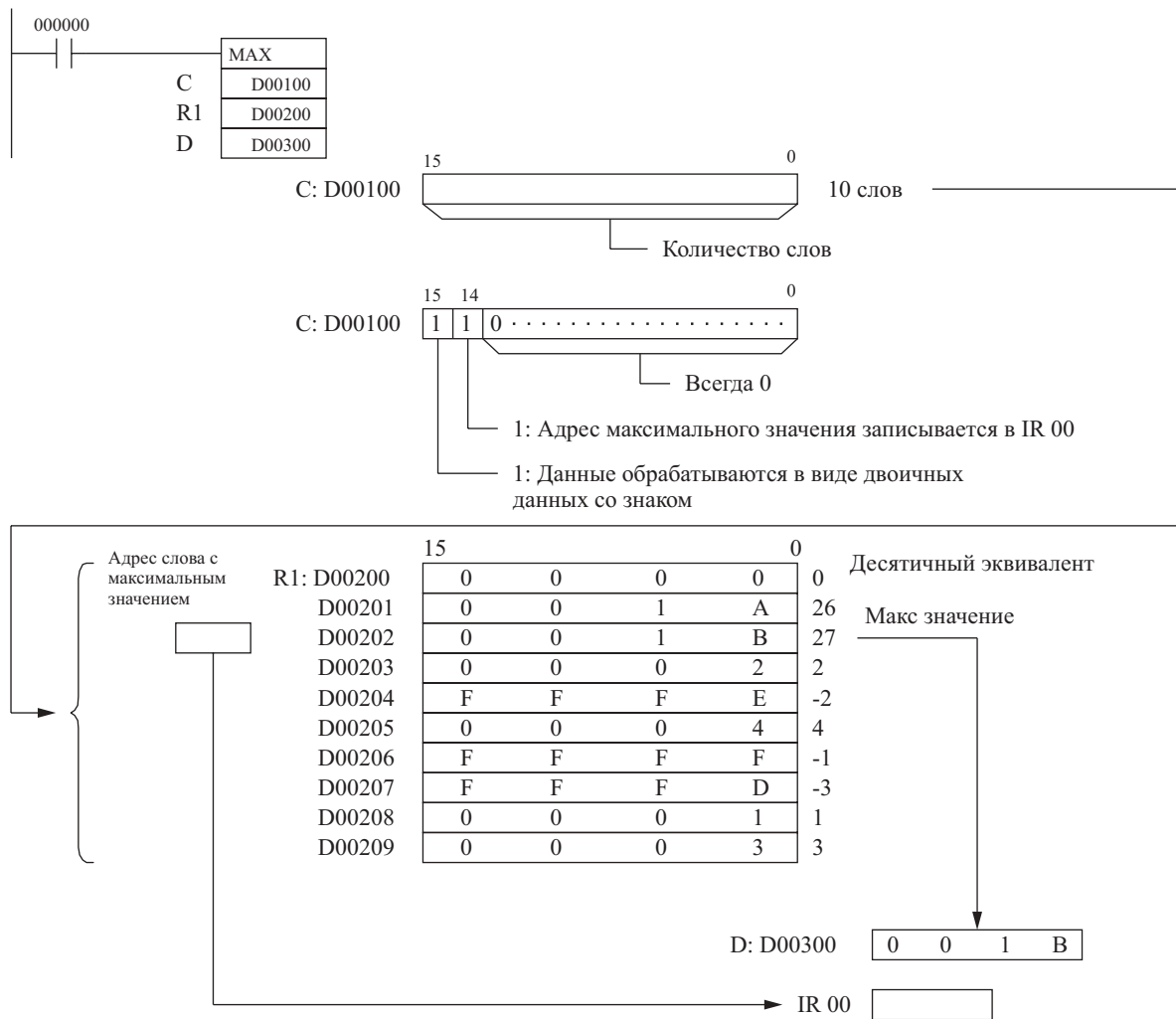
**Меры предосторожности**

Когда бит 15 слова C+1 устанавливается в значение, равное 1, данные в заданном диапазоне обрабатываются как двоичные данные со знаком, а шестнадцатеричные значения от 8000 до FFFF принимаются за отрицательные значения. Таким образом, результаты поиска будут отличаться в зависимости от установки типа данных.

**Примеры**

Когда в следующем ниже примере CIO 000000 находится в состоянии ON, по команде MAX(182) производится просмотр 10 слов, начиная с D00200, с целью поиска максимального значения. Максимальное значение записывается в D00300, а адрес слова, содержащего максимальное значение, записывается в IR00.





### 3-16-11 Команда нахождения максимума FIND MINIMUM: MIN(183)

#### Назначение

По команде MIN(183) производится поиск минимального значения в заданном диапазоне слов.

#### Символ релейно-контактной схемы

MIN(183)	
C	C: Первое управляющее слово
R1	R1: Первое слово в диапазоне слов
D	D: Слово назначения

#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	MIN(183)
	Выполняется один раз для дифференцирования вверх	@ MIN(183)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

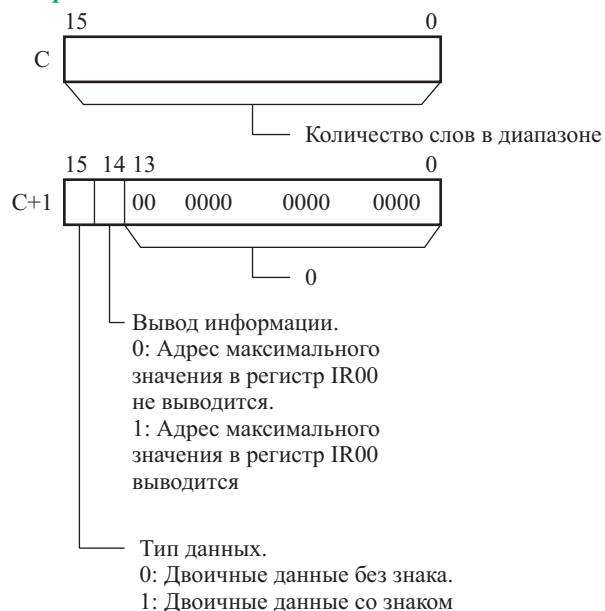
#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****С и С+1: Управляющие слова**

С указывает количество слов в диапазоне. Бит 15 слова С+1 указывает, в каком виде производится обработка данных: в виде двоичных данных со знаком или в виде двоичных данных без знака. Бит 14 слова С+1 указывает, будет ли выводиться в индексный регистр IR00 адрес слова, содержащего минимальное значение.

**Примечание:** Слова С и С+1 должны находиться в одной области данных.

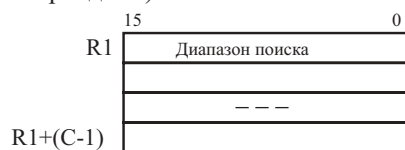


В следующей ниже таблице показаны возможные значения С,

С+1	Тип данных	Запись адреса
0000	Двоичные данные без знака	Нет
4000	Двоичные данные без знака	Да
8000	Двоичные данные со знаком	Нет
С000	Двоичные данные со знаком	Да

**R1: Первое слово в диапазоне слов**

R1 Указывает первое слово диапазона, в котором осуществляется поиск. Слова от R1 до R1+(С-1) просматриваются на предмет поиска минимального значения. (С- это количество слов, задаваемых в операнде С.)



**Примечание:** Слова R1...R1+(С-1) должны находиться в одной области данных.

**Спецификации операндов**

Область	С	R1	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	CIO 0000...CIO 6143	
Рабочая область	W000...W510	W000...W511	
Область удержания бита (Holding bit area)	H000...H510	H000...H511	
Область вспомогательных битов	A000...A958	A000...A959	A448...A959
Область таймера	T0000...T4094	T0000...T4095	
Область счетчика	C0000...C4094	C0000...C4095	
Область DM	D00000...D32766	D00000...D32767	
Область EM, не содержащая банков	E00000...E32766	E00000...E32767	

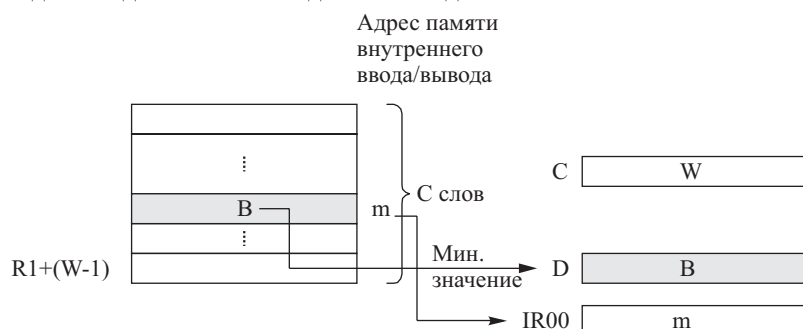
Область	C	R1	D
Область EM, содержащая банки	En_00000...En_32766 (n = 0... C)	En_00000...En_32767 (n = 0... C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)		
Константы	Только указанные значения	–	
Регистры данных	–	DR0...DR15	
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++), ,IR15(++) ,(- -)IR0 ,(- -)IR15		

### Описание

По команде MIN(183) осуществляется просмотр области памяти от R1 до R1+(C-1) и поиск в этом диапазоне слова, содержащего минимальное значение. После нахождения минимального значения оно записывается в D.

Если бит 14 слова C+1 установлен в состояние 1, при выполнении команды MIN(183) адрес слова, содержащего минимальное значение, записывается в индексный регистр IR00. (Если в заданном диапазоне содержится два и более минимальных значений, в индексный регистр записывается адрес первого из слов, содержащих минимальное значение.)

Если бит 15 слова C+1 устанавливается в значение, равное 1, команда MIN(183) обращается с данными в заданном диапазоне как с двоичными данными со знаком.



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание C, находится за пределами значений от 0001 до FFFF. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, если минимальное значение равно 0000. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, если бит 15 слова, содержащего минимальное значение, равен 1. OFF в других случаях.

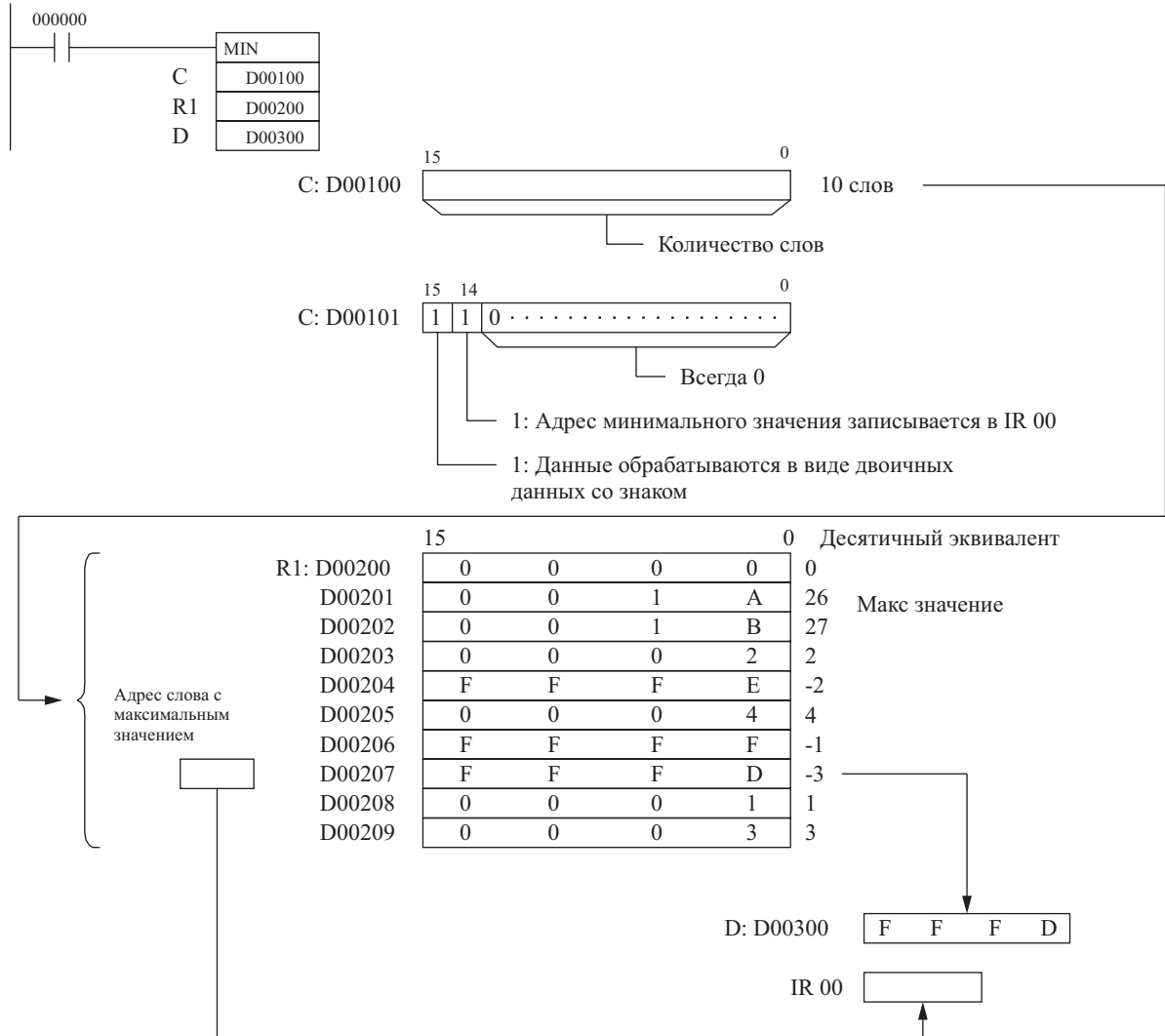
### Меры предосторожности

Когда бит 15 слова C+1 устанавливается в значение, равное 1, данные в заданном диапазоне обрабатываются как двоичные данные со знаком, а шестнадцатеричные значения от 8000 до FFFF принимаются за отрицательные значения. Таким образом, результаты поиска будут отличаться в зависимости от установки типа данных.

### Примеры

Когда в следующем ниже примере CIO 000000 находится в состоянии ON, по команде MIN(183) производится просмотр 10 слов, начиная с D00200, с целью поиска минимального значения.

Минимальное значение записывается в D00300, а адрес слова, содержащего минимальное значение, записывается в IR00.

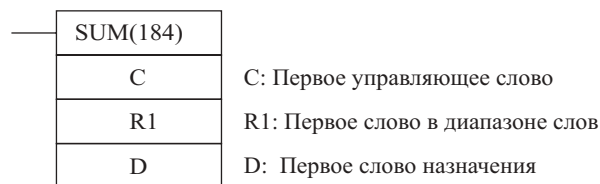


### 3-16-12 Команда вычисления суммы SUM: SUM(184)

#### Назначение

По команде SUM(184) производится прибавление байтов или слов в диапазоне слов и вывод результата в два слова.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	SUM(184)
	Выполняется один раз для дифференцирования вверх	@ SUM(184)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

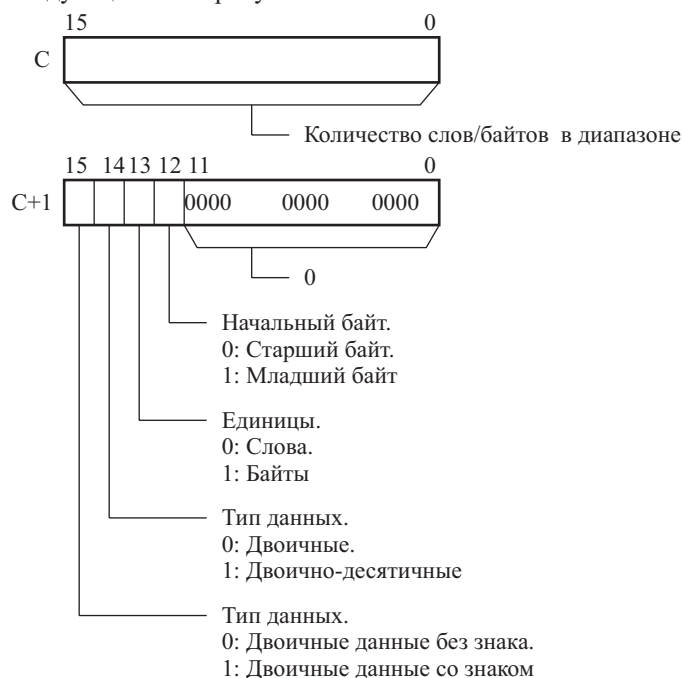
**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****С и С+1: Управляющие слова**

С указывает количество единиц (байтов или слов), подлежащих суммированию. (Бит 13 слова С+1 определяет суммирование байтов или слов.)

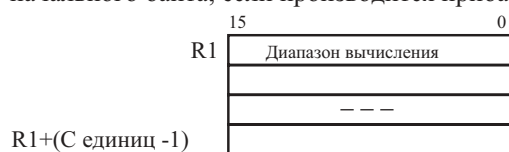
Биты 12 – 15 слова С+1 определяют тип данных при выполнении суммирования, как показано на следующем ниже рисунке.



**Примечание:** Слова С и С+1 должны находиться в одной области данных.

**R1: Первое слово в диапазоне слов**

R1 указывает первое слово диапазона. Длина диапазона зависит как от количества единиц, так и от начального байта, если производится прибавление байтов.



**Примечание:** Все слова в диапазоне выполнения суммирования, должны находиться в одной области данных.

**D: первое слово назначения**

Результат вычисления заносится в слова D+1 и D. Четыре цифры старших разрядов записываются в D+1, а четыре цифры младших разрядов записываются в D.

**Спецификации операндов**

Область	С	R1	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	CIO 0000...CIO 6143	CIO 0000...CIO 6142
Рабочая область	W000...W510	W000...W511	W000...W510
Область удержания бита (Holding bit area)	H000...H510	H000...H511	H000...H510
Область вспомогательных битов	A000...A958	A000...A959	A448...A959
Область таймера	T0000...T4094	T0000...T4095	T0000...T4094
Область счетчика	C0000...C4094	C0000...C4095	C0000...C4094
Область DM	D00000...D32766	D00000...D32767	D00000...D32766

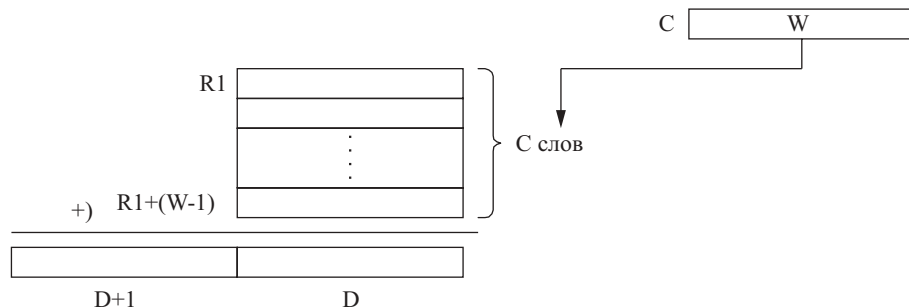
Область	C	R1	D
Область EM, не содержащая банков	E00000...E32766	E00000...E32767	E00000...E32766
Область EM, содержащая банки	En_00000...En_32766 (n = 0... C)	En_00000...En_32767 (n = 0... C)	En_00000...En_32766 (n = 0... C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)		
Константы	Только указанные значения	-	
Регистры данных	-		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++),IR15(++) ,(- -)IR0 ,(- -)IR15		

### Описание

По команде SUM(184) осуществляется прибавление C единиц данных к данным, начиная со слова R1. Результат выводится в D+1 и D. Установки в C+1 определяют тип суммируемых единиц: слова или байты, тип данных: двоичные данные (со знаком или без знака) или двоично-десятичные данные, а также порядок суммирования со старшего байта или с младшего байта слова R1 в том случае, когда задано суммирование байтов.

Когда бит 15 слова C+1 устанавливается в состояние 0, команда SUM(184) обрабатывает двоичные данные. В этом случае бит 12 определяет, являются ли эти данные двоичными данными со знаком (бит 15=1), или без знака (бит 15=0).

Когда бит 13 слова C+1 устанавливается в состояние 1, по команде SUM(184) производится суммирование байтов. В этом случае бит 12 определяет байт, с которого начинается вычисление. Если бит 12 = 1, вычисление начинается со старшего байта, если бит 12 = 0, вычисление начинается с младшего байта.

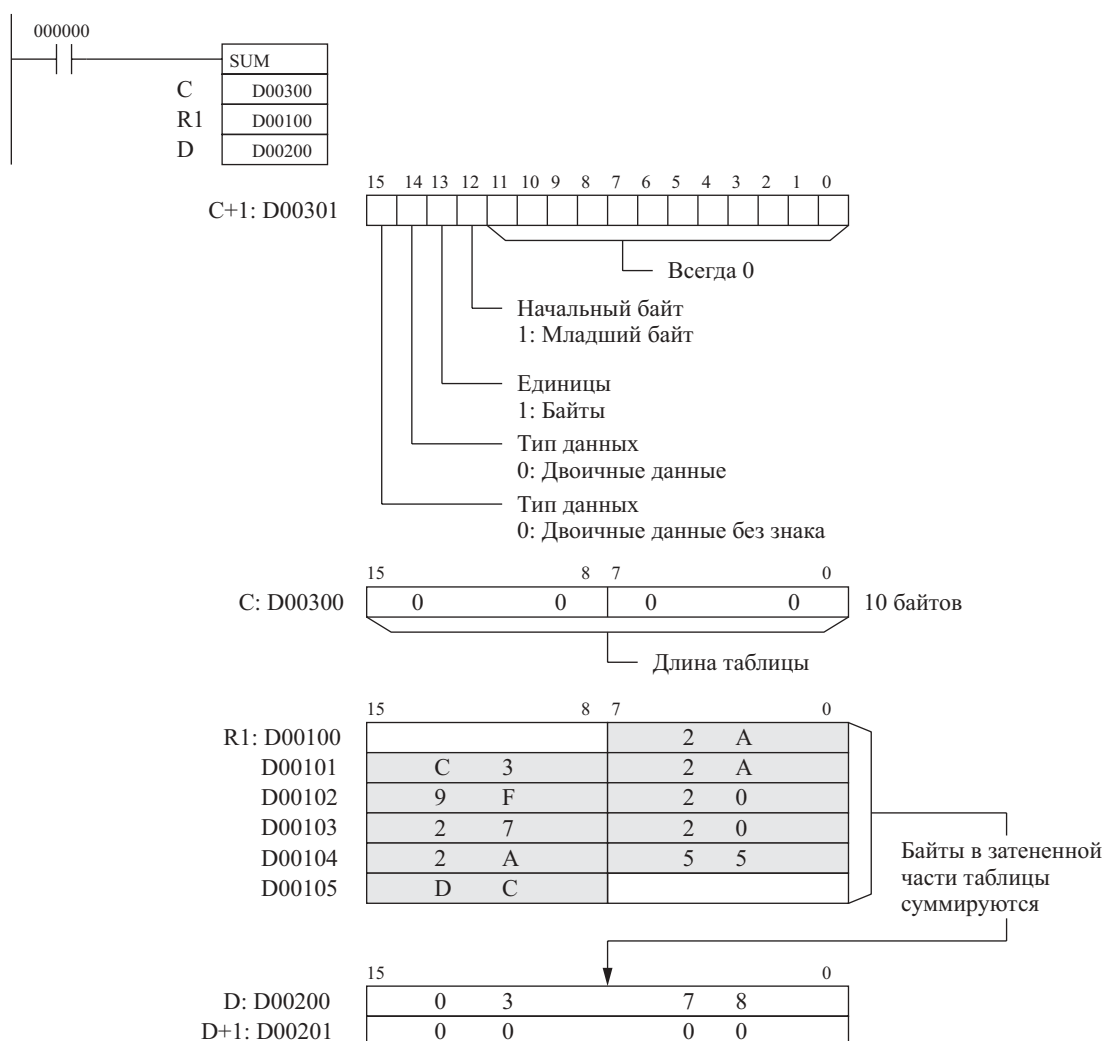


### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание C, находится за пределами значений от 0001 до FFFF. Переводится в состояние ON, если заданы двоично-десятичные данные, однако диапазон содержит двоичные данные. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, если результат равен 0000. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, если бит 15 результирующего слова равен 1. OFF в других случаях.

### Примеры

Когда в следующем ниже примере CIO 000000 находится в состоянии ON, по команде SUM(184) производится прибавление 10 байтов двоичных данных без знака, начиная с младшего байта D00100. Результат записывается в D00201 и D00200.



### 3-16-13 Команда вычисления контрольной суммы кадра FRAME CHECKSUM: FCS(180)

#### Назначение

По команде FCS(180) производится вычисление контрольной суммы кадра в заданном диапазоне слов.

#### Символ релейно-контактной схемы

FCS(180)	
C	C: Первое управляющее слово
R1	R1: Первое слово в диапазоне слов
D	D: Первое слово назначения

#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	FCS(180)
	Выполняется один раз для дифференцирования вверх	@ FCS(180)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

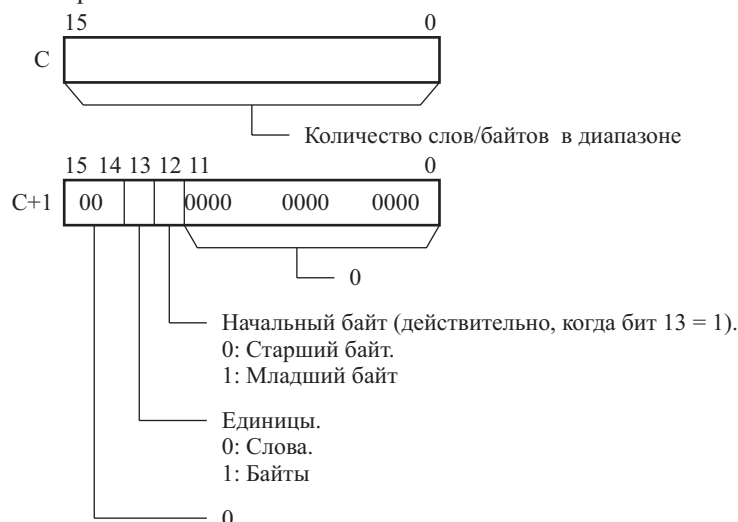
#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****С и С+1: Управляющие слова**

С указывает количество единиц (байтов или слов), используемых при подсчете контрольной суммы. (Бит 13 слова С+1 определяет суммирование байтов или слов.)

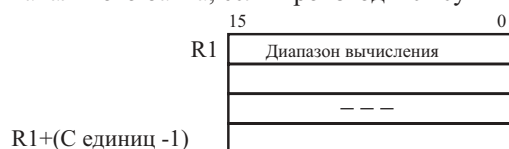
Когда бит 13 слова С+1 устанавливается в состояние 1, по команде FCS(180) вычисляется значение контрольной суммы для байтов данных. В этом случае бит 12 определяет порядок вычисления. Если бит 12=1, вычисление начинается с младшего байта слова R1, если бит 12 = 0, вычисление начинается со старшего байта слова R1.



**Примечание:** Слова С и С+1 должны находиться в одной области данных.

**R1: Первое слово в диапазоне слов**

R1 Указывает первое слово диапазона. Длина диапазона зависит как от количества единиц, так и от начального байта, если производится суммирование байтов.



**Примечание:** Все подлежащие суммированию слова, должны находиться в одной области данных.

**D: первое слово назначения**

Результат вычисления заносится в слово D, если для вычисления выбираются байты.

Результат вычисления заносится в слова D+1 и D, если для вычисления задаются слова. В этом случае четыре цифры старших разрядов записываются в D+1, а четыре цифры младших разрядов записываются в D.

**Спецификации операндов**

Область	С	R1	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	CIO 0000...CIO 6143	
Рабочая область	W000...W510	W000...W511	
Область удержания бита (Holding bit area)	H000...H510	H000...H511	
Область вспомогательных битов	A000...A958	A000...A959	A448...A959
Область таймера	T0000...T4094	T0000...T4095	
Область счетчика	C0000...C4094	C0000...C4095	
Область DM	D00000...D32766	D00000...D32767	
Область EM, не содержащая банков	E00000...E32766	E00000...E32767	
Область EM, содержащая банки	En 00000...En_32766 (n = 0... C)	En 00000...En_32767 (n = 0... C)	En 00000...En_32766 (n = 0... C)

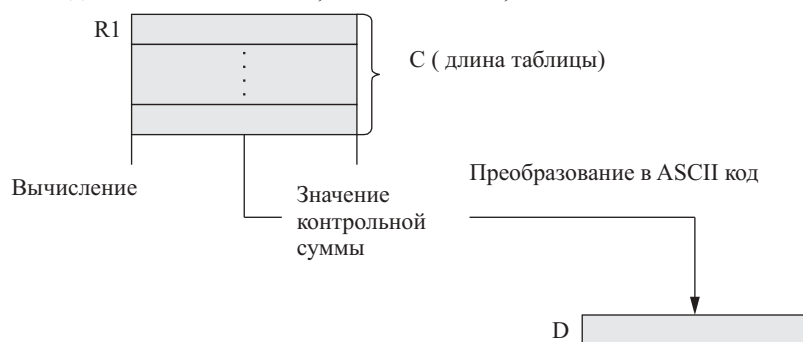


Область	C	R1	D
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0... C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0... C)		
Константы	Только указанные значения	–	
Регистры данных	–		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++) ,IR15+(++) ,(- -)IR0 ,(- -)IR15		

### Описание

По команде FCS(180) осуществляется вычисление контрольной суммы для C единиц данных, начиная с данных в слове R1. При этом производится преобразование данных в ASCII код и вывод результата в D (для байтов) или в D+1 и D (для слов). Установки в C+1 определяют тип суммируемых единиц (слова или байты), тип данных (двоичные данные со знаком или без знака, или двоично-десятичные данные), а также порядок выполнения суммирования (со старшего байта или с младшего байта слова R1) в том случае, когда задано суммирование байтов.

Когда бит 13 слова C+1 устанавливается в состояние 1, команда FCS(180) оперирует байтами данных. В этом случае бит 12 определяет начальный байт при вычислениях. Если бит 12 = 1, вычисления начинаются с младшего байта слова R1, если бит 12 = 0, вычисления начинаются со старшего байта слова R1.

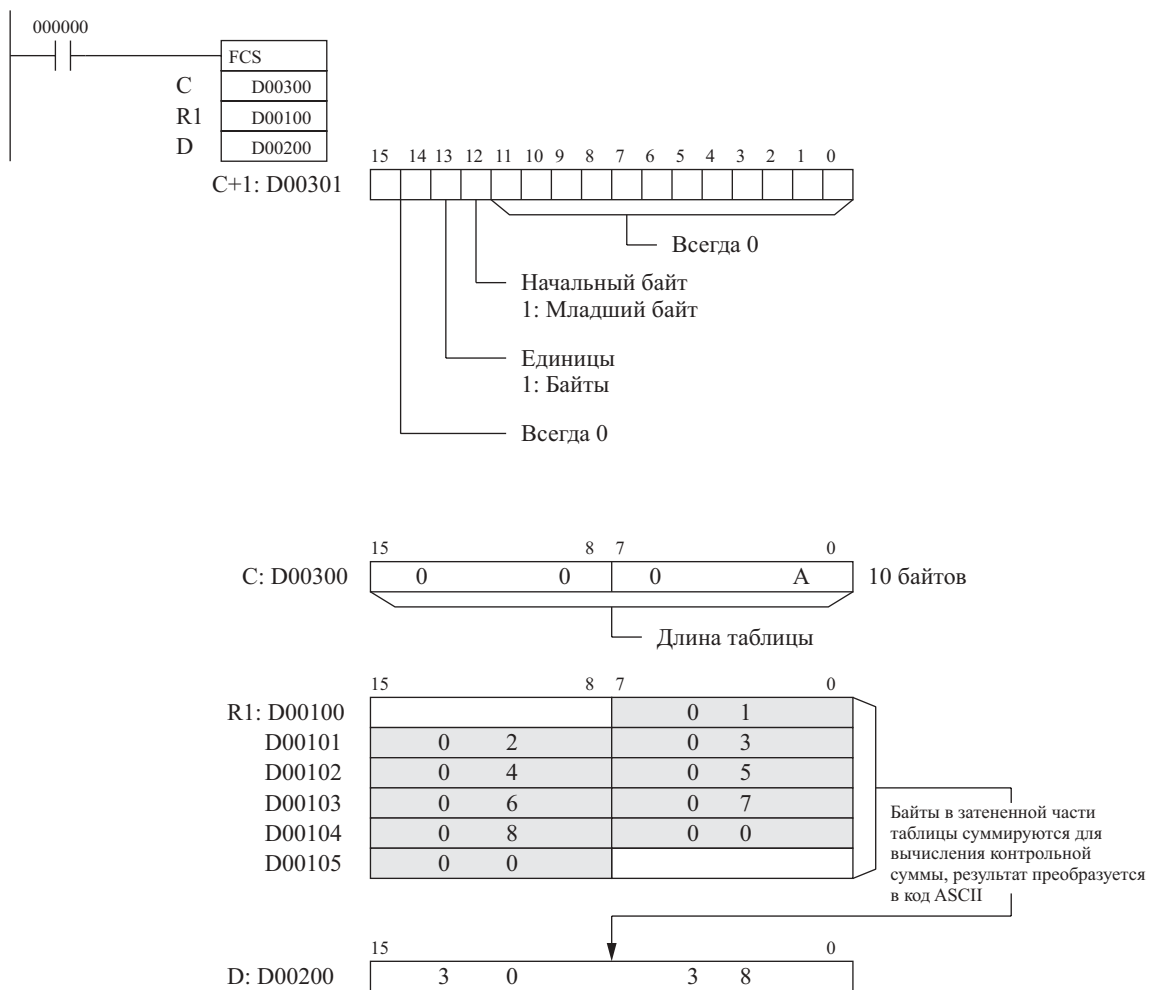


### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание C, находится за пределами значений от 0001 до FFFF. OFF в других случаях.

### Примеры

Когда в следующем ниже примере CIO 000000 находится в состоянии ON, по команде FCS(180) производится вычисление значения контрольной суммы для 10 байтов данных, начиная с младшего байта D00100. Результат записывается в D00200.



## 3-17 Команды управления данными

## 3-17-1 Команда PID-регулирования PID CONTROL: PID(190)

**Назначение**

По команде PID(190) осуществляется выполнение PID регулирования согласно заданным параметрам.

**Символ релейно-контактной схемы**

PID(190)	
S	S: Входное слово
C	C: Слово первого параметра
D	D: Выходное слово

**Модификации**

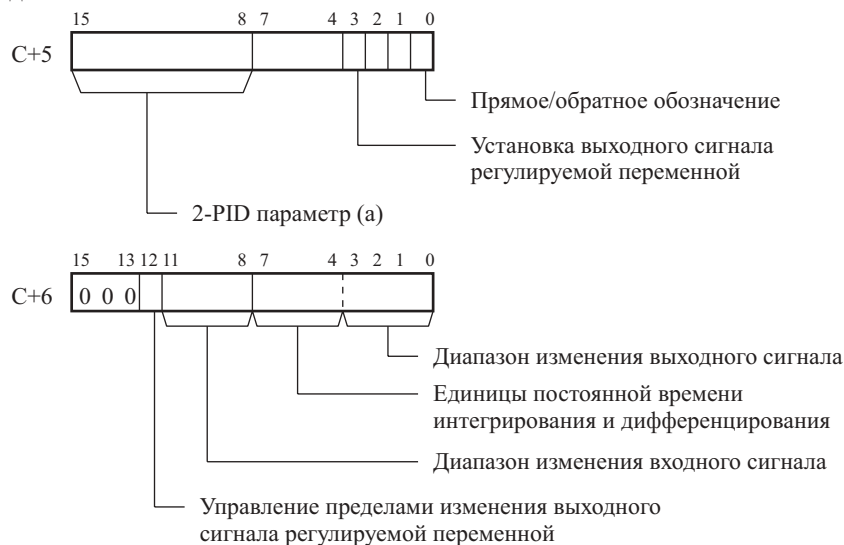
Модификации	Выполняется в каждом цикле для состояния ON	PID(190)
	Выполняется один раз для дифференцирования вверх	@ PID(190)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не применяется	Да	Да	Не применяется

**Параметры**

На следующем ниже рисунке показано расположение данных параметров. Для детального ознакомления с параметрами обратитесь к параграфу «Установки параметров PID регулирования» далее в настоящем разделе.

**Спецификации операндов**

Область	S	C	D
Область ввода/вывода (область CIO)	CIO...0000...CIO 6143	CIO 0000...CIO 6105	CIO 0000...CIO 6143
Рабочая область	W000...W511	W000...W473	W000...W511
Область удержания бита (Holding bit area)	H000...H511	H000...H473	H000...H511
Область вспомогательных битов	A000...A959	A000...A921	A448...A959
Область таймера	T0000...T4095	T0000...T4057	T0000...T4095
Область счетчика	C0000...C4094	C0000...C4057	C0000...C4095
Область DM	D00000...D32767	D00000...D32729	D00000...D32767
Область EM, не содержащая банков	E00000...E32767	E00000...E32729	E00000...E32767

Область	S	C	D
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32729 (n = 0...C)	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	DR0...DR15	–	DR0...DR15
Регистры данных	–		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,-( - )IR15		

### Описание

Когда условие выполнения находится в состоянии ON, по команде PID(190) осуществляется вычисление конечной величины с двумя степенями свободы согласно параметрам, указанным в C (заданное значение, постоянные PID регулирования, и т.д.). При этом берутся двоичные данные в указанном диапазоне из содержания входного слова S, и выполняется PID- регулирование согласно заданным параметрам. Результат сохраняется в виде регулируемой переменной в выходном слове D.

Параметры применяются, когда состояние выполнения переводится из состояния OFF в состояние ON. Флаг ошибки переводится в состояние ON, когда установки находятся за пределами допустимых значений.

Если установки находятся в допустимых пределах, PID регулирование производится с использованием начальных значений. В этом случае плавное регулирование не выполняется. Оно будет использоваться для регулируемых переменных при последующем выполнении PID регулирования. (Плавное регулирование – это регулирование, при котором изменение регулируемой переменной производится постепенно и непрерывно, с целью предотвращения неблагоприятного эффекта вследствие внезапных изменений.)

Когда условие выполнения переводится в состояние ON, вводится текущее значение для указанного периода выборки и начинается выполнение регулирования.



Число битов входных данных в пределах 16-ти битов текущего значения (S) определяется установкой диапазона в слове C+6 (биты от 08 до 11). Например, если для диапазона входных значений задано 12 битов (4 шестн.), числа от 0000 до 0FFF (шестн.) будут являться текущими значениями. (Числа, большие, чем 0FFF, будут приравняться значению 0FFF.)

Диапазон задаваемых значений также зависит от диапазона входных значений.

Измеряемые значения (текущие значения) и заданные значения выражены двоичными числами без знака от 0000 до максимального значения входного диапазона.

Число битов выходных данных в пределах 16-ти битов регулируемой переменной определяется установкой диапазона в слове C+6 (биты от 00 до 03). Например, если для диапазона выходных значений задано 12 битов (4 шестн.), регулируемая переменная выводится в виде значений от 0000 до 0FFF (шестн.).

При работе только в режиме пропорционального регулирования вывод регулируемой переменной в том случае, когда текущее значение равняется заданному значению, может устанавливаться либо равным 0%, либо 50%.

Пропорциональное регулирование может задаваться либо в прямом направлении, либо в обратном направлении.

Верхний и нижний пределы выходных значений регулируемой переменной может устанавливаться пользователем.

Дискретизация выборки может устанавливаться в единицах, равных 10 мсек. (от 0.01 до 99.00 сек.), однако в действительности PID регулирование будет определяться комбинацией периодичности выборки и времени выполнения команды PID(190) (в каждом из циклов).

Из всех параметров PID регулирования (С...С+8), только заданное значение может изменяться после перевода условия выполнения в состояние ON. При необходимости изменения других параметров, непременно измените состояние условия выполнения из ON в OFF.

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание С, находится за пределами допустимых значений. Переводится в состояние ON, если действительный период выборки данных вдвое превышает указанный период выборки данных. OFF в других случаях.
Флаг «Более чем»	>	Переводится в состояние ON, если после выполнения PID регулирования значение регулируемой переменной превышает верхний предел. OFF в других случаях.
Флаг «Менее чем»	<	Переводится в состояние ON, если после выполнения PID регулирования значение регулируемой переменной ниже нижнего предела. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON в процессе выполнения PID регулирования. OFF в других случаях.

#### Меры предосторожности

Выполнение команды PID(190) производится, как если бы условием ее выполнения являлся сигнал «STOP-RUN». Вычисления PID регулирования производятся, когда условие выполнения остается в состоянии ON в следующем цикле после инициализации параметров С+9...С+38. Поэтому, при использовании флага «Всегда ON» в качестве условия выполнения команды PID(190), при запуске всегда осуществляйте отдельный процесс инициализации С+9...С+38.

Если данные в слове С находятся за пределами допустимых значений, определяется ошибка и флаг ошибки переводится в состояние ON.

В случае, когда действительный период выборки данных более чем вдвое превышает указанный период выборки данных, определяется ошибка и флаг ошибки переводится в состояние ON. Тем не менее, в этом случае выполнение PID регулирования продолжается.

В процессе выполнения PID регулирования Флаг переноса переводится в состояние ON.

Флаг «Более чем» переводится в состояние ON, когда после выполнения PID регулирования значение регулируемой переменной превышает верхний предел. В этом случае выводится результирующее значение, равное верхнему пределу.

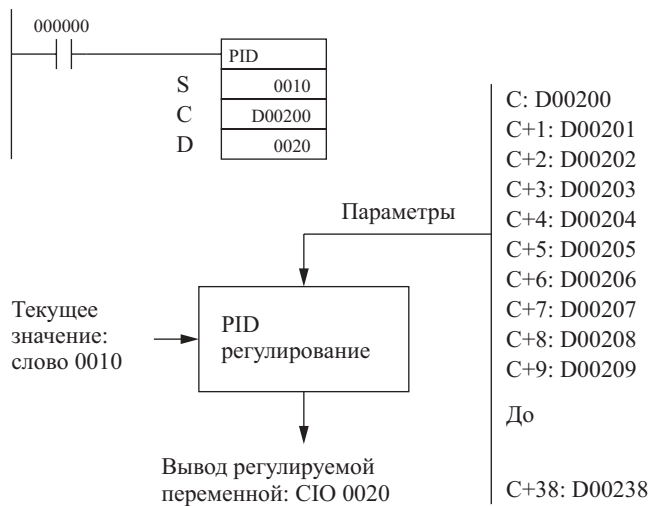
Флаг «Менее чем» переводится в состояние ON, когда после выполнения PID регулирования значение регулируемой переменной меньше нижнего предела. В этом случае выводится результирующее значение, равное нижнему пределу.

Из всех параметров PID регулирования (С...С+38), только заданное значение в слове С может изменяться после перевода условия выполнения в состояние ON. После изменения других параметров для ввода в действие нового значения параметра непременно измените состояние условия выполнения из OFF в ON.

#### Примеры

При возрастающем изменении условия выполнения СЮ 000000 (из OFF в ON), рабочая область в D00209...D00238 подвергается инициализации согласно параметрам, заданным в D00200...D00208. После инициализации рабочей области осуществляется выполнение PID регулирования и вывод регулируемой переменной в СЮ 0020.

Когда СЮ 000000 переводится в состояние ON, PID регулирование выполняется с указанной периодичностью выборки данных, согласно параметрам, заданным в D00200...D00208. Значение регулируемой переменной выводится в СЮ 00200.



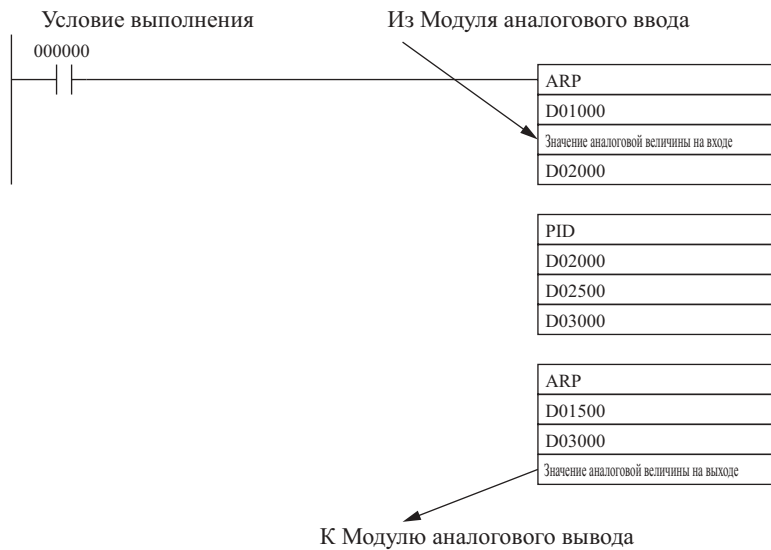
### Значения входных величин и диапазон значений регулируемой переменной

Число битов входных данных для измеряемого (текущего) значения определяется установкой диапазона входных величин в C+6 (биты 08...11), а число битов выходных данных для регулируемой переменной определяется установкой диапазона выходных значений в C+6 (биты 0...3). В следующей ниже таблице приводятся эти диапазоны.

C+6, биты 08...11, или C+6, биты 00...03	Количество действительных битов	Диапазон
0	8	0000...00FF шестн.
1	9	0000...01FFшестн.
2	10	0000...03FF шестн.
3	11	0000...07FF шестн.
4	12	0000...0FFF шестн.
5	13	0000...1FFF шестн.
6	14	0000...3FFF шестн.
7	15	0000...7FFF шестн.
8	16	0000...FFFF шестн.

Если диапазон данных, обрабатываемых Модулем аналогового ввода, или Модулем аналогового вывода, не может точно определяться количеством битов, для преобразования данных в соответствующие значения можно применять команду APR(069) (ARITHMETIC PROCESS) до выполнения команды PID(160) и после ее выполнения.

Следующий ниже раздел программы показывает пример использования Модуля аналогового ввода DRT1-AD04 и Модуля аналогового вывода DRT1-DA02 в качестве Slave-модулей CompoBus/D. Диапазон величин для этих двух Модулей – от 0000 до 1770 (шестн.) не может устанавливаться простым указанием действительного количества цифр. Поэтому для преобразования диапазона входных величин для Модуля аналогового модуля от 0000 до 1770 в диапазон от 0000 до FFFF применяется команда APR(069). Для вывода данных из Модуля аналогового вывода после выполнения команды PID(190) выходные величины преобразуются в диапазон от 0000 до 1770 (шестн.) при помощи повторной команды APR(069).



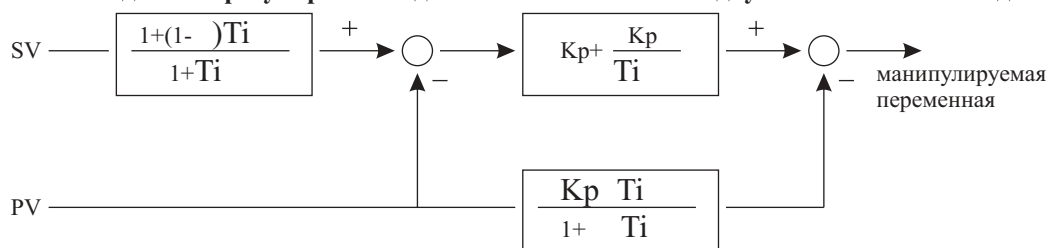
**Характеристики**

Наименование		Характеристики	
Метод выполнения PID регулирования		–	PID регулирование до конечного значения, фильтрового типа, две степени свободы (вперед/назад).
Количество петель PID регулирования		–	Не ограничено (одна петля для каждой из команд).
Периодичность выборки		$\tau$	0.01...99.99 сек
Константы PID регулирования	Пропорциональный диапазон	P	0.1...999.9 %
	Постоянная времени интегрирования	Tik	1...8191, 9999 (Интегральное регулирование не выполняется для периода выборки 9999.)
	Постоянная времени дифференцирования	Tdk	0...8191, (Дифференциальное регулирование не выполняется для периода выборки 0.)
Заданное значение		SV	0...65535 (Действительно до максимального входного значения.)
Измеряемое значение		PV	0...65535 (Действительно до максимального входного значения.)
Регулируемая переменная		MV	0...65535 (Действительно до максимального входного значения.)

**Метод вычисления**

Вычисление при PID регулировании выполняется по конечному значению с фильтрацией и двумя степенями свободы.

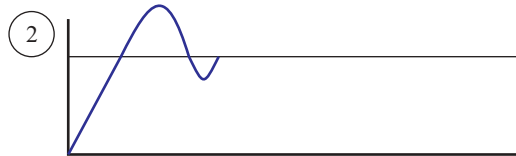
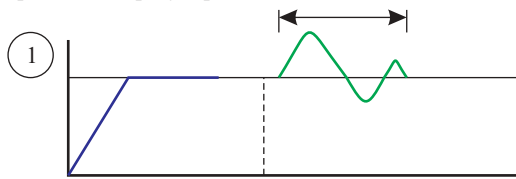
**Блок схема для PID регулирования до конечного значения с двумя степенями свободы**



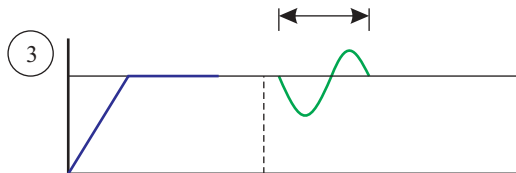
- Kp: постоянная пропорционального регулирования.
- Ti: постоянная времени интегрирования.
- Td: постоянная времени дифференцирования.
- $\tau$ : периодичность выборки данных.
- $\alpha$ : Параметр 2-PID регулирования.
- $\lambda$ : Коэффициент неполного дифференцирования.

Когда предотвращение перерегулирования осуществляется при помощи простого PID регулирования, стабилизация возмущений замедляется (1). С другой стороны, когда стабилизация возмущений ускоряется, возникает перерегулирование и достижение конечного значения замедляется (2).

При использовании PID регулирования по конечному значению с двумя степенями свободы перерегулирование не возникает, а стабилизация возмущений и достижение конечного значения ускоряются (3).  
Простое PID регулирование



PID регулирование по конечному значению



#### Установки параметров PID регулирования

Данные регулирования	Наименование	Содержание	Пределы установки
C	Заданное значение	Контролируется конечное значение процесса.	Двоичные данные (такое же количество битов, которое задано для входного диапазона.)
C+1	Пропорциональный диапазон	Параметр, определяющий диапазон пропорционального регулирования/общий диапазон регулирования.	0001...270F шестн. (от 1 до 9999); (от 0.1% до 999.9% с дискретностью 0.1%).
C+2	Tik	Константа, выражающая эффективность интегрирующего действия. По мере увеличения данного параметра интегрирующее действие уменьшается.	0001...1FFF шестн. (от 1 до 8191); (9999 – интегральное регулирование не выполняется). (См. примечание 1.)
C+3	Tdk	Константа, выражающая эффективность дифференцирующего действия. По мере увеличения данного параметра дифференцирующее действие уменьшается.	0001...1FFF шестн. (от 1 до 8191); (0000 дифференциальное регулирование не выполняется). (См. примечание 1.)
C+4	Периодичность выборки данных ( $\tau$ )	Устанавливает период для выполнения PID регулирования.	0001...270F шестн. (от 1 до 9999); (от 0.01 до 99.99 сек с дискретностью 10 мсек.).
Биты от 04 до 15 слова C+5	2-PID параметр ( $\alpha$ )	Коэффициент, определяющий свойства входного фильтра. В обычном случае используйте значение, равное 0.65 (т.е. установку 000). По мере приближения параметра к значению, равному нулю, эффективность фильтра уменьшается.	000 шестн.: $\alpha=0.65$ . Установки от 100 до 163 шестн. Означает, что значение двух цифр после запятой изменяется от 0.00 до 0.99 (См. примечание 2.)
Бит 03 слова C+5	Задание вывода данных регулируемой переменной	Задает выходное значение регулируемой переменной, когда текущее значение равно заданному значению.	0: Выход равен 0%. 1: Выход равен 50%.
Бит 00 слова C+5	Указание направления PID регулирования (вперед/назад)	Определяет направление регулирования при пропорциональном действии.	0: Действие в обратном направлении. 1: Действие в прямом направлении.



Данные регулирования	Наименование	Содержание	Пределы установки
Бит 12 слова C+6	Контроль пределов значения выходных данных регулируемой переменной	Устанавливает или отменяет контроль пределов изменения выходных данных регулируемой переменной.	0: Отключен (контроль пределов не выполняется). 1: Включен (контроль пределов выполняется).
Биты от 08 до 11 слова C+6	Диапазон входных значений	Определяет количество битов входных данных.	0: 8 битов, 1: 9 битов, 2: 10 битов, 3: 11 битов, 4: 12 битов, 5: 13 битов, 6: 14 битов, 7: 15 битов, 8: 16 битов.
Биты от 04 до 07 слова C+6	Диапазон выходных значений	Определяет количество битов выходных данных. (Количество битов выходных данных автоматически устанавливается равным количеству битов входных данных).	
Биты от 00 до 03 слова C+6	Единицы постоянной времени интегрирования/дифференцирования	Определяет единицы для выражения постоянной времени интегрирования и постоянной времени дифференцирования.	1: Множитель периода выборки. 9: Время (единицы: 100 мсек.)
C + 7	Нижний предел выходного значения регулируемой переменной	Нижний предел для случая, когда устанавливается нижний предел выходного значения регулируемой переменной.	0000 FFFF (двоичное) (См. примечание 3.)
C+ 8	Верхний предел выходного значения регулируемой переменной	Верхний предел для случая, когда устанавливается верхний предел выходного значения регулируемой переменной.	0000 FFFF (двоичное) (См. примечание 3.)

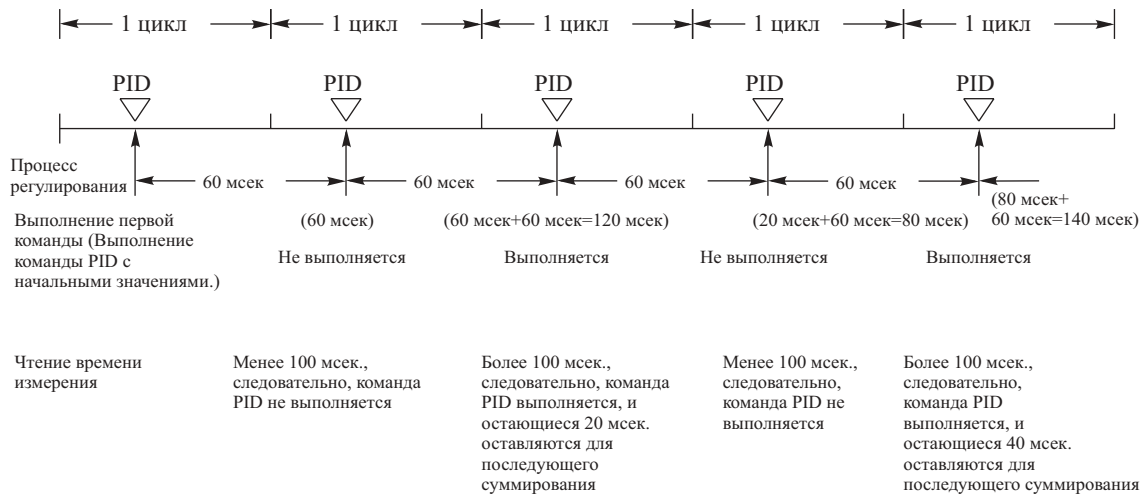
- Примечание:**
1. Когда для единиц (в C+6) выбирается значение 1, диапазон значений параметра устанавливается от 1 до 8191 длительности цикла. Когда для единиц (в C+6) выбирается значение 9, диапазон значений параметра - от 0.1 до 819.1 сек. Когда для единиц выбирается значение 9, устанавливайте постоянную времени интегрирования и постоянную времени дифференцирования в значение, входящее в диапазон от 1 до 8191 длительности интервала между выборками.
  2. Установка параметра 2-PID ( ) в значение, равное 000, означает 0.65, т.е. обычное значение.
  3. При включении функции контроля предела выходного значения регулируемой переменной (т.е. установки параметра в значение 1) устанавливайте значения следующим ниже образом.

#### Периодичность выборки данных и длительность цикла

Период выборки данных может устанавливаться в пределах от 0.01 до 99.99 сек. в единицах, равных 10 мсек. Тем не менее, реальное выполнение PID регулирования зависит от комбинации значений периода выборки данных и времени выполнения команды (в каждом из циклов). Соотношения между длительностью периода выборки данных и длительностью цикла являются следующими:

- Когда длительность выборки меньше длительности цикла, PID регулирование выполняется в каждом цикле, но не в каждой выборке.
- Когда длительность выборки данных превышает длительность периода или равна длительности цикла, PID регулирование выполняется не в каждом цикле, однако команда PID(190) выполняется, когда суммарная длительность циклов (т.е. время между командами PID регулирования) становится равной или превышает длительность выборки. Излишек, остающийся от суммарного времени (т.е. суммарная длительность нескольких циклов минус длительность выборки), суммируется с последующими несколькими циклами.

Например, предположим, что длительность выборки равна 100 мсек, а длительность цикла постоянна и равна 60 мсек. В первом цикле после начала выполнения команда PID(190) выполняться не будет, так как длительность цикла меньше длительности выборки. Что касается второго цикла, к моменту его окончания суммарная длительность составляет 60 + 60 мсек., что превышает длительность выборки (100 мсек.), поэтому команда PID(190) выполняется. Остаток, равный 20 мсек. (т.е. 120 – 100 = 20 мсек.) суммируется со следующим циклом. Процедура повторяется в последующих циклах.



### Действия, выполняемые при регулировании

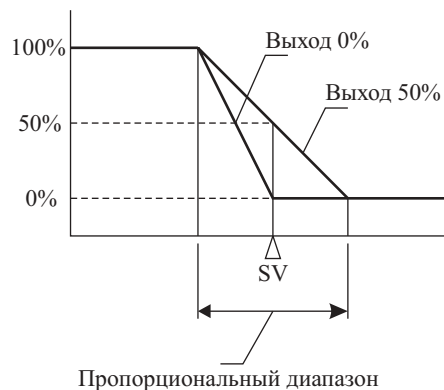
#### Пропорциональное регулирование (P)

Пропорциональное регулирование – это регулирование, при котором пропорциональный диапазон указывается относительно заданного значения (SV) и в этом диапазоне величина регулируемой переменной пропорциональна отклонению (от заданного значения, прим переводчика). На следующем ниже рисунке приводится пример для регулирования в обратном направлении.

В случае, когда при выполнении пропорционального регулирования текущее значение (PV) становится меньше нижнего значения пропорционального диапазона, величина регулируемой переменной становится равной 100% (т.е. равной максимальному значению). В пределах пропорционального диапазона величина регулируемой переменной пропорциональна отклонению (т.е. разнице между заданным значением и текущим значением), и постепенно уменьшается до тех пор, пока текущее значение не станет равным заданному значению (т.е. до тех пор, пока отклонение не станет равным нулю). В этот момент величина регулируемой переменной становится равной 0%, (или 50 % в зависимости от установки параметра, определяющего величину регулируемой переменной). Величина регулируемой переменной будет равна 0%, когда текущее значение больше заданного значения.

Ширина пропорционального диапазона выражается в процентах от общего диапазона изменения входного сигнала. Чем меньше ширина пропорционального диапазона, тем больше постоянная пропорционального регулирования и сильнее корректирующее воздействие. При пропорциональном регулировании обычно остается некоторое смещение (остаточное отклонение), однако это смещение можно уменьшить путем задания меньшей ширины пропорционального диапазона. Однако, если ширина пропорционального диапазона слишком мала, при выполнении регулирования возникают броски перерегулирования.

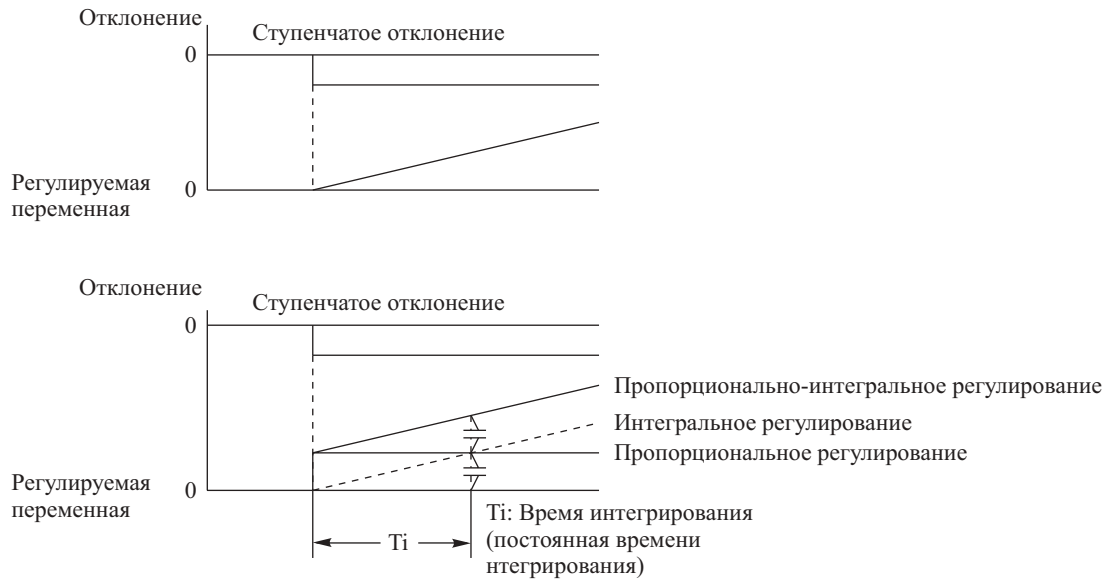
Пропорциональное регулирование (обратное направление)



#### Интегральное регулирование (I)

Комбинация интегрального регулирования с пропорциональным регулированием позволяет с истечением времени уменьшить смещение, обеспечивая равенство текущего значения заданному значению. Эффективность воздействия при интегральном регулировании определяется временем интегрирования, которое равно времени, требуемому регулируемой переменной для достижения значения, которое она достигает при пропорциональном регулировании при появлении ступенчатого отклонения, как показано на следующем рисунке. Чем меньше время интегрирования, тем сильнее воздействие при выполнении интегрально-

го регулирования. Если время интегрирования слишком мало, воздействие будет слишком сильным, что вызовет появление перерегулирования.

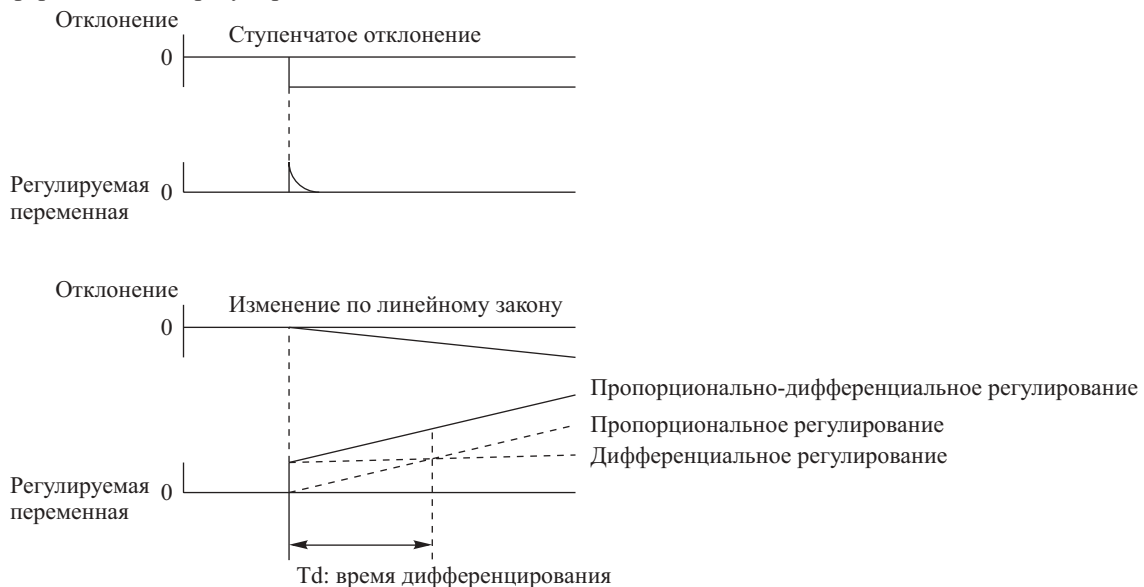


#### Дифференциальное регулирование (D)

Пропорциональное регулирование и интегральное регулирование осуществляют регулирование контролируемой величины, поэтому задержка результирующего изменения неизбежна. Дифференциальное регулирование осуществляет компенсацию этого недостатка.

В ответ на стремительное изменение контролируемой величины при таком регулировании производится большое изменение регулируемой переменной, осуществляя быстрое восстановление первоначального состояния. Коррекция регулируемой переменной выполняется пропорционально наклону (коэффициенту дифференцирования), вызванному отклонением измеряемой величины.

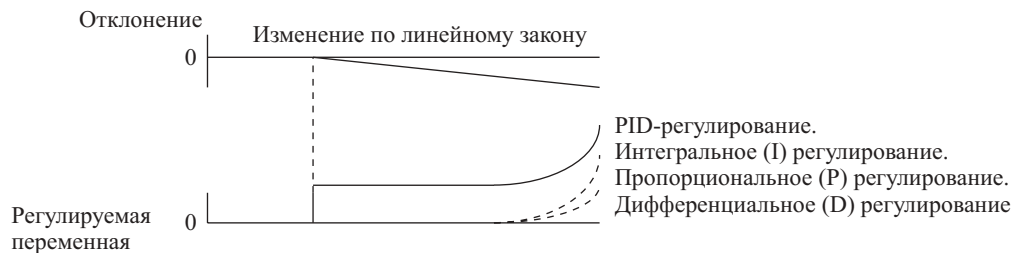
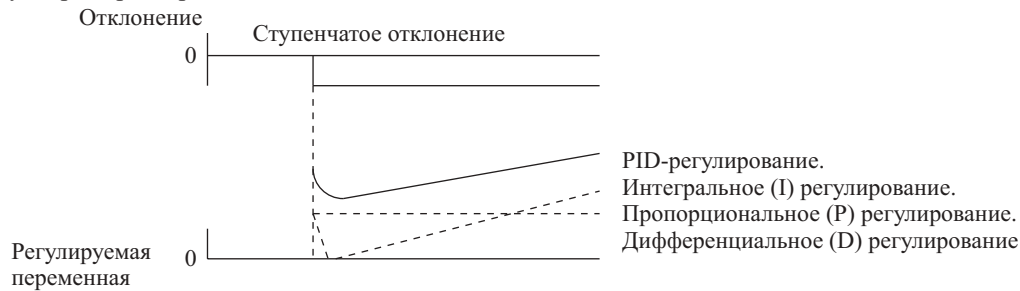
Эффективность воздействия при дифференциальном регулировании определяется временем интегрирования, которое равно времени, требуемому регулируемой переменной для достижения значения, которое она достигает при пропорциональном регулировании при появлении ступенчатого отклонения, как показано на следующем рисунке. Чем больше время дифференцирования, тем сильнее воздействие при дифференциальном регулировании.



#### PID регулирование

PID регулирование объединяет пропорциональное регулирование (P), интегральное регулирование (I) и дифференциальное регулирование (D). Такое регулирование обеспечивает превосходные результаты даже при управлении объектами, обладающими существенным временем запаздывания. Пропорциональная часть обеспечивает плавное регулирование, характеризующееся отсутствием скачков перерегулиро-

вания, интегральная часть автоматически устраняет любое смещение, а дифференциальная часть ускоряет реагирование системы на возникающие отклонения.

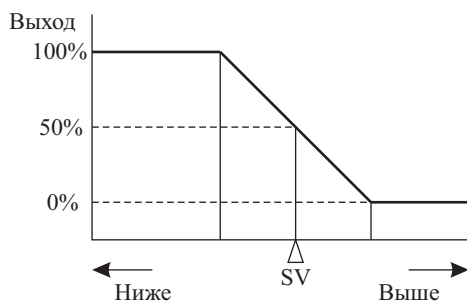


### Направление регулирования

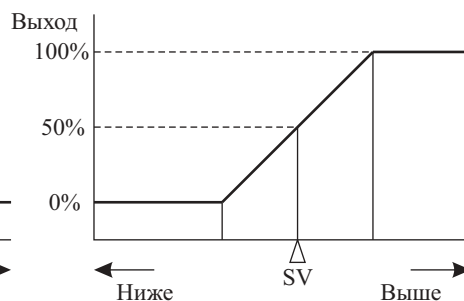
При использовании PID - регулирования выберите одно из направлений регулирования. В одном из направлений регулируемая переменная увеличивается с увеличением разницы между заданным значением и текущим значением.

- Прямое направление: регулируемая переменная увеличивается, когда текущее значение больше заданного значения.
- Противоположное (обратное) направление: регулируемая переменная увеличивается, когда текущее значение меньше заданного значения.

Обратное направление



Прямое направление



### Регулировка PID параметров

Основные соотношения между PID параметрами и состоянием регулирования показаны ниже.

- Когда процесс допускает наличие некоторого времени стабилизации, однако не допускает возникновения перерегулирования, увеличивайте ширину пропорционального диапазона.
- Когда процесс допускает наличие перерегулирования, однако, желательна быстрая стабилизация процесса, уменьшайте ширину пропорционального диапазона. Однако, когда ширина пропорционального диапазона слишком мала, возможно возникновение рывка.
- Когда при выполнении регулирования возникает значительное рывкание, или когда при регулировании текущее значение не может достигать заданного значения вследствие перерегулирования, вероятной причиной является повышенная эффективность интегральной составляющей регулирования. В этом случае перерегулирование будет уменьшаться при увеличении постоянной времени интегрирования или увеличении ширины пропорционального диапазона.
- Когда длительность цикла мала и возникает рывкание, вероятной причиной может быть высокое быстроедействие системы и повышенная эффективность дифференциальной составляющей регулирования. В этом случае уменьшайте эффективность дифференциального регулирования.

## 3-17-2 Команда контроля пределов выходных данных LIMIT CONTROL: LMT(680)

**Назначение**

По команде LMT(680) осуществляется управление выходными данными в зависимости от того, находятся ли входные данные в установленных пределах, нижнем и верхнем.

**Символ релейно-контактной схемы**

LMT(680)	
S	S: Входное слово
C	C: Первое слово предела
D	D: Выходное слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	LMT(680)
	Выполняется один раз для дифференцирования вверх	@ LMT(680)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	S	C	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6142	CIO 0000...CIO 6143
Рабочая область	W000...W511	W000...W510	W000...W511
Область удержания бита (Holding bit area)	H000...H511	H000...H510	H000...H511
Область вспомогательных битов	A000...A959	A000...A958	A448...A959
Область таймера	T0000...T4095	T0000...T4094	T0000...T4095
Область счетчика	C0000...C4095	C0000...C4094	C0000...C4095
Область DM	D00000...D32767	D00000...D32766	D00000...D32767
Область EM, не содержащая банков	E00000...E32767	E00000...E32729	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32729 (n = 0...C)	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF (двоичные)	–	
Регистры данных	DR0...DR15	–	DR0...DR15
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-)IR15		

**Описание**

Когда условие выполнения находится в состоянии ON, по команде LMT(680) осуществляется управление выходными данными в зависимости от нахождения заданных входных данных (двоичных данных со знаком, 16 битов) в установленных пределах - нижнем и верхнем. Содержание слов C и C+1 приводится в следующей таблице.

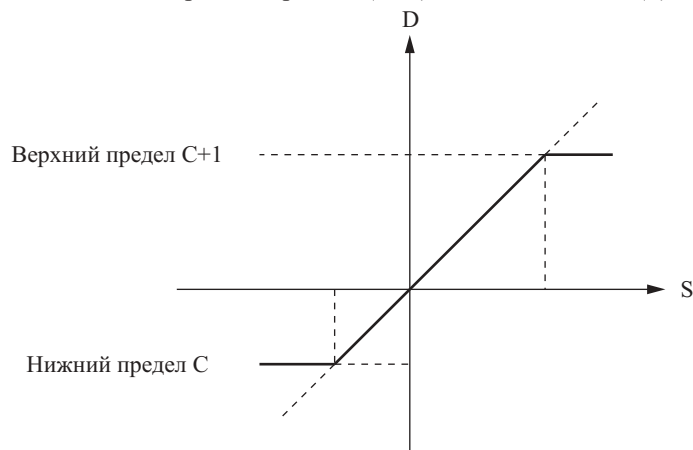
C	Данные нижнего предела (минимальное значение выходных данных)
C+1	Данные верхнего предела (максимальное значение выходных данных)

C и C+1 должны находиться в одной области данных.

Если значение входного слова (S) меньше нижнего предела (C), данные нижнего предела записываются в D, а флаг «Менее чем» переводится в состояние ON.

Если значение входного слова (S) больше верхнего предела (C+1), данные верхнего предела записываются в D, а флаг «Более чем» переводится в состояние ON.

Если значение входного слова (S) больше или равно значению нижнего предела (C), или меньше или равно значению верхнего предела (C+1), входные данные (S) записываются в D.



#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение верхнего предела меньше значения нижнего предела. OFF в других случаях.
Флаг «Более чем»	>	Переводится в состояние ON, когда значение входных данных (S) больше верхнего предела. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, если результат равен нулю.
Флаг «Менее чем»	<	Переводится в состояние ON, когда значение входных данных (S) меньше нижнего предела. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, если бит старшего разряда результата равен 1. OFF в других случаях.

#### Меры предосторожности

Если значение верхнего предела меньше значения нижнего предела определяется ошибка и флаг ошибки переводится в состояние ON.

Когда значение входных данных (S) больше верхнего предела, флаг «Более чем» переводится в состояние ON.

Если выходное слово D равно 0000 (шестн.) флаг равенства переводится в состояние ON.

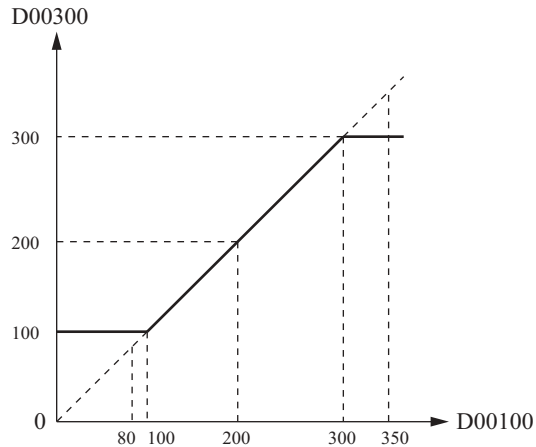
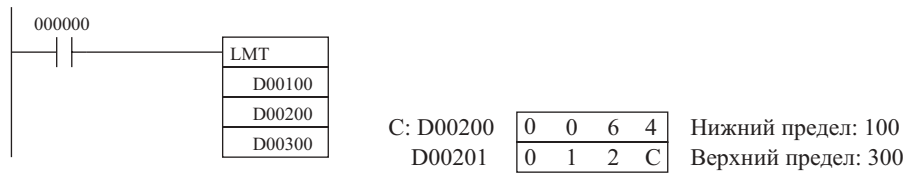
Когда значение входных данных (S) меньше нижнего предела флаг «Менее чем» переводится в состояние ON.

#### Пример

Если значение D00100 равно 0050 шестн. (80), в D00300 записывается значение 0064 шестн. (100), так как 80 меньше нижнего предела, равного 100.

Если значение D00100 равно 00C8 шестн. (200), в D00300 записывается значение 00C8 шестн. (200), так как 200 находится между верхним и нижним пределом.

Если значение D00100 равно 015E шестн. (350), в D00300 записывается значение 012C шестн. (300), так как 350 больше верхнего предела, равного 300.

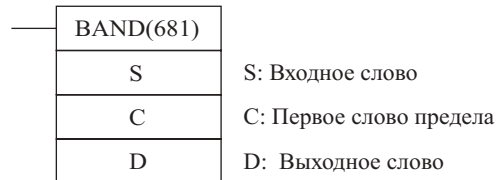


### 3-17-3 Команда контроля пределов зоны нечувствительности DEAD BAND CONTROL: BAND(681)

#### Назначение

По команде BAND(681) осуществляется управление выходными данными в зависимости от того, находятся ли входные данные в зоне нечувствительности.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	BAND(681)
	Выполняется один раз для дифференцирования вверх	@ BAND(681)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Спецификации операндов

Область	S	C	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6142	CIO 0000...CIO 6143
Рабочая область	W000...W511	W000...W510	W000...W511
Область удержания бита (Holding bit area)	H000...H511	H000...H510	H000...H511
Область вспомогательных битов	A000...A959	A000...A958	A448...A959
Область таймера	T0000...T4095	T0000...T4094	T0000...T4095
Область счетчика	C0000...C4095	C0000...C4094	C0000...C4095
Область DM	D00000...D32767	D00000...D32766	D00000...D32767

Область	S	C	D
Область EM, не содержащая банков	E00000...E32767	E00000...E32729	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32729 (n = 0...C)	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF (двоичные)	–	
Регистры данных	DR0...DR15	–	DR0...DR15
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-)IR15		

### Описание

Когда условие выполнения находится в состоянии ON, по команде BAND(681) осуществляется управление выходными данными в зависимости от нахождения указанных входных данных (двоичных данных со знаком, 16 битов) в пределах зоны нечувствительности - нижнем и верхнем. Содержание слов C и C+1 приводится в следующей таблице.

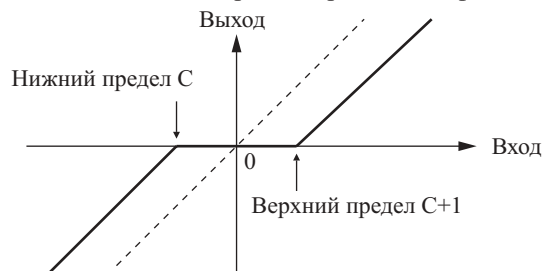
C	Данные нижнего предела (нижний предел зоны нечувствительности).
C+1	Данные верхнего предела (верхний предел зоны нечувствительности).

C и C+1 должны находиться в одной области данных.

Если значение входного слова (S) больше или равно значению нижнего предела (C), или меньше или равно значению верхнего предела (C+1), в D записывается значение 0000 (шестн.), и флаг равенства переводится в состояние ON.

Если значение входного слова (S) меньше нижнего предела (C), в D записывается разница между входными данными и нижним пределом, а флаг «Менее чем» переводится в состояние ON.

Если значение входного слова (S) больше верхнего предела (C+1), в D записывается разница между входными данными и верхним пределом, а флаг «Более чем» переводится в состояние ON.



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение верхнего предела меньше значения нижнего предела. OFF в других случаях.
Флаг «Более чем»	>	Переводится в состояние ON, когда значение входных данных (S) больше верхнего предела. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, если результат равен нулю.
Флаг «Менее чем»	<	Переводится в состояние ON, когда значение входных данных (S) меньше нижнего предела. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, если бит старшего разряда результата равен 1. OFF в других случаях.



**Меры предосторожности**

Если значение верхнего предела меньше значения нижнего предела определяется ошибка и флаг ошибки переводится в состояние ON.

Когда значение входных данных (S) больше верхнего предела, флаг «Более чем» переводится в состояние ON.

Если выходное слово D равно 0000 шестн. Флаг равенства переводится в состояние ON.

Когда значение входных данных (S) меньше нижнего предела флаг «Менее чем» переводится в состояние ON.

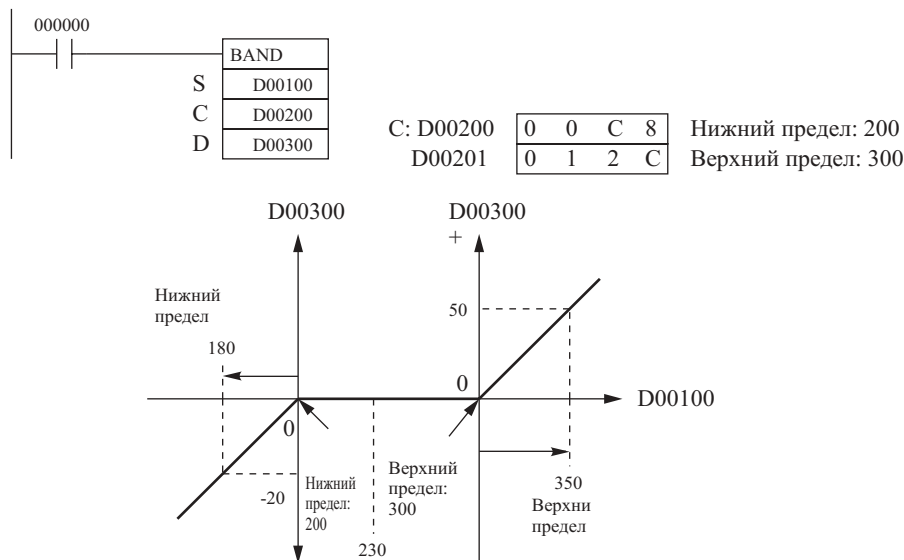
Когда бит старшего разряда выходного слова равен 1, флаг отрицательного значения переводится в состояние ON.

**Пример**

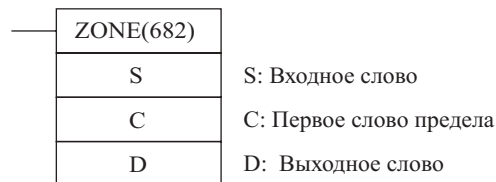
Если значение D00100 равно 00B4 шестн. (180), в D00300 записывается значение  $180 - 200 = \text{FFEC}$  0064 шестн. (-20), так как 180 меньше нижнего предела, равного 200.

Если значение D00100 равно 00E6 шестн. (230), в D00300 записывается значение 0, так как 230 находится между верхним и нижним пределом.

Если значение D00100 равно 015E шестн. (350), в D00300 записывается значение  $350 - 300 = 0032$  шестн. (50), так как 350 больше верхнего предела, равного 300.

**3-17-4 Команда контроля зоны нечувствительности DEAD ZONE CONTROL: ZONE(682)****Назначение**

По команде ZONE(682) осуществляется прибавление заданного смещения к входным данным и вывод результата.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	ZONE(682)
	Выполняется один раз для дифференцирования вверх	@ ZONE(682)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Спецификации операндов**

Область	S	C	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6142	CIO 0000...CIO 6143
Рабочая область	W000...W511	W000...W510	W000...W511
Область удержания бита (Holding bit area)	H000...H511	H000...H510	H000...H511
Область вспомогательных битов	A000...A959	A000...A958	A448...A959
Область таймера	T0000...T4095	T0000...T4094	T0000...T4095
Область счетчика	C0000...C4095	C0000...C4094	C0000...C4095
Область DM	D00000...D32767	D00000...D32766	D00000...D32767
Область EM, не содержащая банков	E00000...E32767	E00000...E32729	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32729 (n = 0...C)	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF (двоичные)	–	
Регистры данных	DR0...DR15	–	DR0...DR15
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-( - )IR15		

**Описание**

Когда условие выполнения находится в состоянии ON, по команде ZONE(682) производится прибавление заданного смещения к указанным входным данным (двоичные данные со знаком, 16 битов) и вывод результата в указанное слово. Содержание слов C и C+1 приводится в следующей таблице.

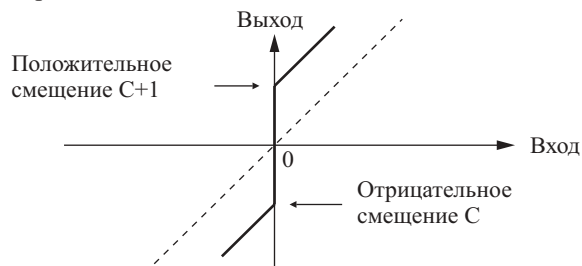
C	Отрицательное смещение.
C+1	Положительное смещение.

C и C+1 должны находиться в одной области данных.

Если значение входного слова (S) меньше нуля, в D записывается сумма, состоящая из входных данных и отрицательного смещения, а флаг «Менее чем» переводится в состояние ON.

Если значение входного слова (S) больше нуля, в D записывается сумма, состоящая из входных данных и положительного смещения, а флаг «Более чем» переводится в состояние ON.

Если значение входного слова (S) равно нулю, в D записывается значение 0000 (шестн.), и флаг равенства переводится в состояние ON.



В случае, когда выходные данные меньше, чем 8000 (шестн.) или больше, чем 7FFF, знак изменяется на противоположный. Например, для отрицательного смещения, равного FF00 (шестн.) и входных данных, равных 8000 (шестн.) выходные данные будут равны:

$$8000 \text{ (шестн.) } [ - 32768 ] - FF00 \text{ (шестн.) } [ - 256 ] = 7F00 \text{ (шестн.) } [ 32512 ]$$

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение верхнего предела меньше значения нижнего предела. OFF в других случаях.
Флаг «Более чем»	>	Переводится в состояние ON, когда значение входных данных (S) больше верхнего предела. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, если результат равен нулю.
Флаг «Менее чем»	<	Переводится в состояние ON, когда значение входных данных (S) меньше нижнего предела. OFF в других случаях.
Флаг отрицательного значения	N	Переводится в состояние ON, если бит старшего разряда результата равен 1. OFF в других случаях.

**Меры предосторожности**

Если значение верхнего предела меньше значения нижнего предела определяется ошибка и флаг ошибки переводится в состояние ON.

Когда значение входных данных (S) больше верхнего предела, флаг «Более чем» переводится в состояние ON.

Если выходное слово D равно 0000 шестн. Флаг равенства переводится в состояние ON.

Когда значение входных данных (S) меньше нижнего предела флаг «Менее чем» переводится в состояние ON.

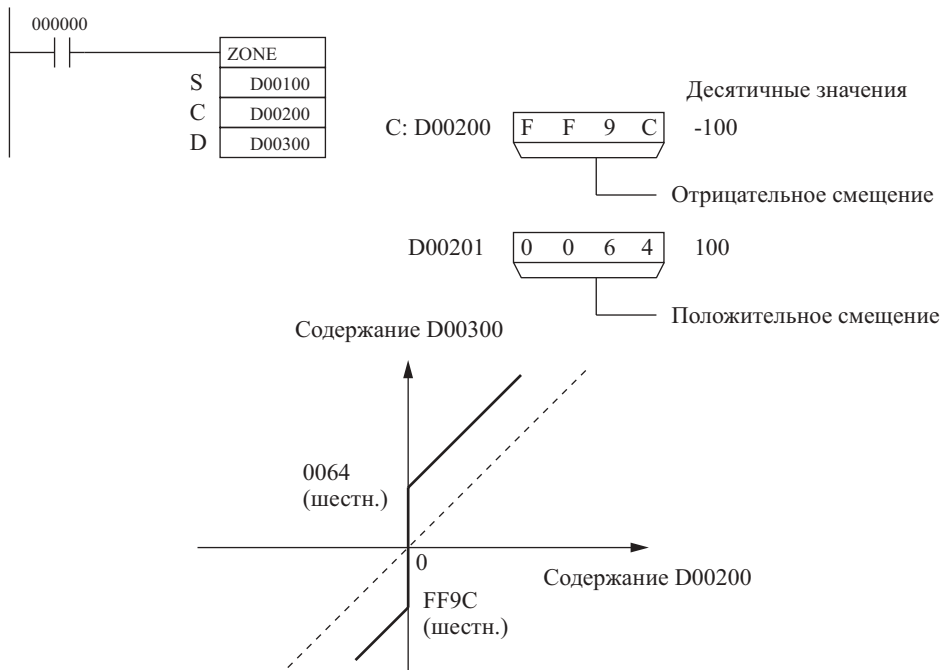
Когда бит старшего разряда выходного слова равен 1, флаг отрицательного значения переводится в состояние ON.

**Пример**

Когда СЮ 000000 находится в состоянии ON, смещение, равное -100 прибавляется к значению D00100, если это значение меньше 0, и результирующее значение записывается в D00300.

Если значение D00100 равно 0, в D00300 записывается значение 0000 (шестн.).

Если значение D больше нуля, прибавляется смещение, равное +100, и результирующая величина записывается в D00300.



## 3-17-5 Команда преобразования данных SCALING: SLC(194)

**Назначение**

По команде SLC(194) производится преобразование двоичных данных без знака в двоично-десятичные данные без знака, соответственно указанной линейной зависимости.

**Символ релейно-контактной схемы**

—	SCL(194)	
—	S	S: Исходное слово
—	P1	P1: Слово первого параметра
—	R	R: Результирующее слово

**Модификации**

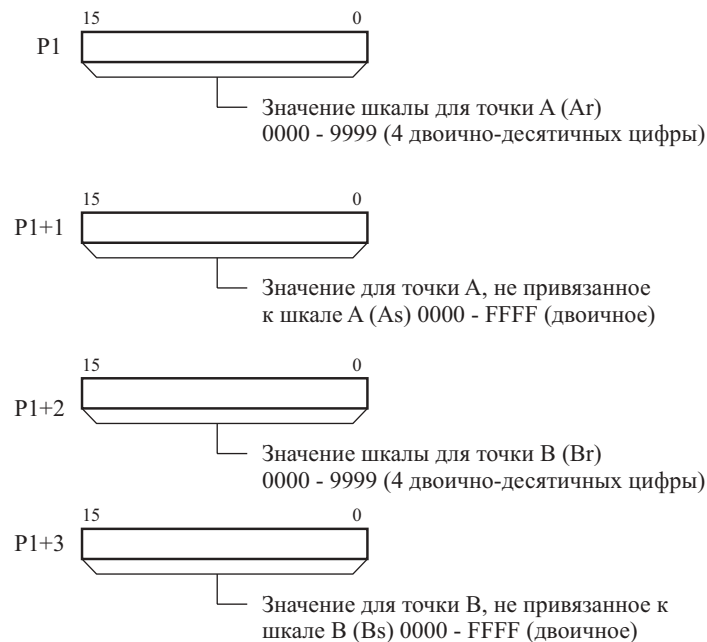
Модификации	Выполняется в каждом цикле для состояния ON	SLC(194)
	Выполняется один раз для дифференцирования вверх	@ SLC(194)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды**

Содержание четырех слов, начиная со слова первого параметра (P1) показано на следующем ниже рисунке.

**Спецификации операндов**

Область	S	C	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6140	CIO 0000...CIO 6143
Рабочая область	W000...W511	W000...W508	W000...W511
Область удержания бита (Holding bit area)	H000...H511	H000...H508	H000...H511
Область вспомогательных битов	A000...A959	A000...A956	A448...A959
Область таймера	T0000...T4095	T0000...T4094	T0000...T4095
Область счетчика	C0000...C4095	C0000...C4092	C0000...C4095
Область DM	D00000...D32767	D00000...D32764	D00000...D32767

Область	S	C	D
Область EM, не содержащая банков	E00000...E32767	E00000...E32764	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32764 (n = 0...C)	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	-		
Регистры данных	DR0...DR15	-	DR0...DR15
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-( -)IR15		

### Описание

Команда SCL(194) применяется для преобразования двоичных данных без знака, содержащихся в исходном слове S, в двоично-десятичные данные без знака, и вывода результата в слово R согласно линейной зависимости, определяемой точками (As, Ad) и (Bs, Bd). Адрес слов, содержащих координаты точек (As, Ad) и (Bs, Bd), указывается словом первого параметра P1. Эти точки определяются двумя значениями (As и Bs) перед установкой масштаба и двумя значениями (Ag и Br) после установки масштаба.

Для преобразования используются следующие ниже уравнения.

$$R = Bd - \frac{Bd - Ad}{Bs - As} \times (Bs - As)$$

(Bs-As) - в двоично-десятичном коде.

Крутизна (наклон) характеристики равна

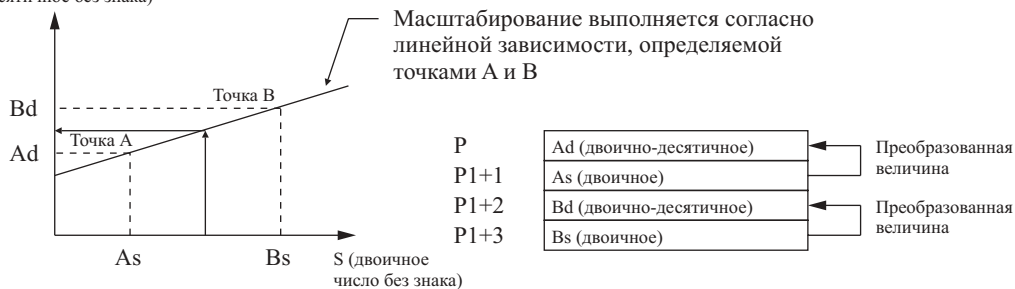
$$R = Bd - \frac{Bd - Ad}{Bs - As}$$

(Bs-As) - в двоично-десятичном коде.

Точки A и B могут определять линейную зависимость с положительной или отрицательной крутизной. Применение отрицательного значения крутизны обеспечивает обратное масштабирование.

Результат округляется до ближайшего целого числа. Если результат меньше нуля, в качестве результата выводится значение, равное 0000. Если результат больше 9999, в качестве результата выводится значение, равное 9999.

R (двоично-десятичное без знака)



Команда SCL(194) может использоваться для преобразования выходных сигналов Модулей аналогового ввода в величину, определяемую пользователем. Например, если входные напряжения от 1 до 5 В, подаваемые на вход Модуля аналогового ввода, вводятся в память как 0000...0FA0 (шестн.), при помощи команды SCL(194) эти значения могут быть преобразованы в значения от 50 до 200С.

По команде SCL(194) производится преобразование двоичных данных без знака в двоично-десятичные данные без знака. Для преобразования отрицательных значений необходимо в программе перед использованием команды SCL(194) прибавить максимальное отрицательное значение к преобразуемому значению (см. пример.).

Команда SCL(194) не может выводить в результирующее слово отрицательное значение. Если результатом является отрицательное число, в результирующее слово выводится значение, равное 0000.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение C (Ar) или C+2 (Br) выражено не в двоично-десятичном коде. Переводится в состояние ON, когда значение C+1 (As) или C+3 (Bs) выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, если результат равен нулю. OFF в других случаях.

**Меры предосторожности**

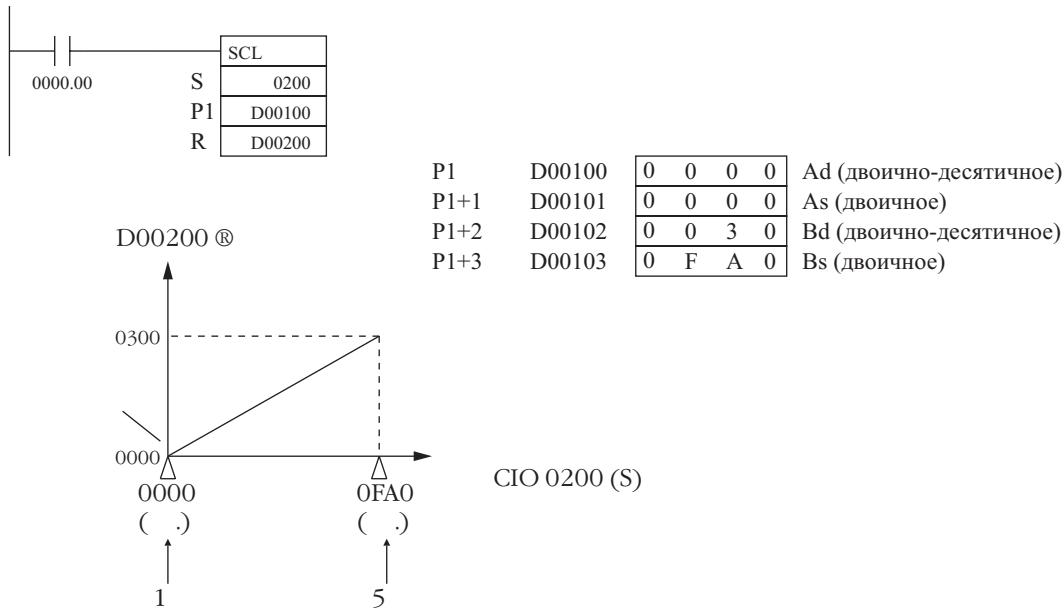
В случае, когда значения Ar (C) или Br (C+2) выражены не в двоично-десятичном коде, или значения As (C+1) и Bs (C+3) равны, определяется ошибка и флаг ошибки переводится в состояние ON.

Если содержание результирующего слова равно 0000, флаг равенства переводится в состояние ON.

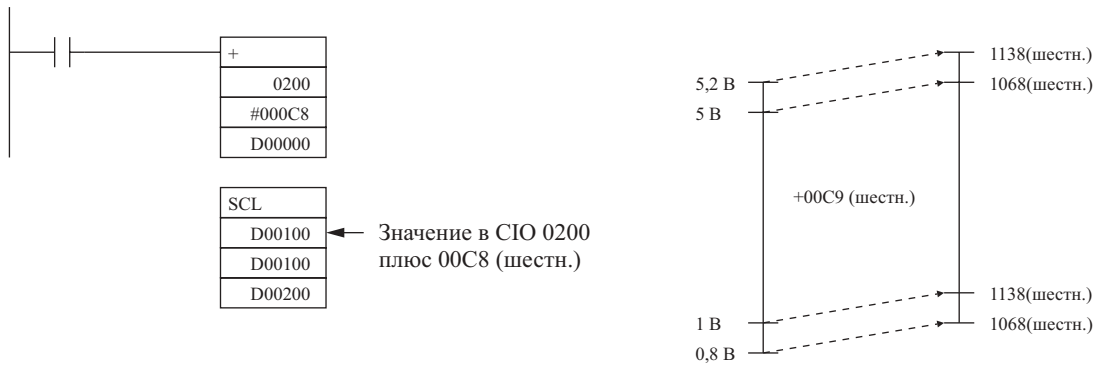
**Примеры**

В следующем ниже примере предполагается, что аналоговый сигнал размахом от 1 до 5 В преобразуется и заносится в CIO 0200 в виде значений от 0000 до 0FA0 (шестн.) Команда SCL (194) используется для преобразования (масштабирования) содержания CIO 0200 в значения от 0000 до 0300 в двоично-десятичном коде.

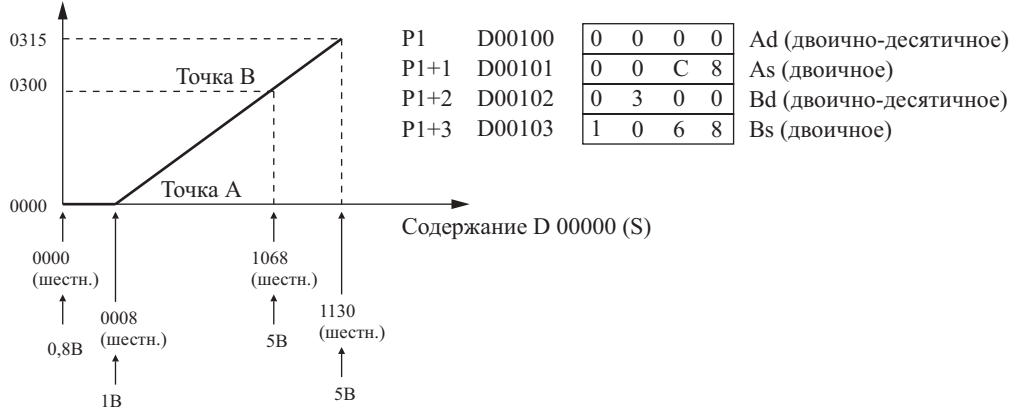
Когда CIO 000000 находится в состоянии ON, содержание CIO 0200 масштабируется с использованием линейной зависимости, определяемой точкой A (0000 0000) и точкой B (0FA0, 0300). Координаты этих точек содержатся в D00100...D00103. Результат выводится в D00200.

**Отрицательные значения**

Для входных значений от 0.8 до 5.2 В, Модуль аналогового ввода в действительности обрабатывает входные величины от FF38 до 1068. Команда SCL(194) тем не менее, может обрабатывать только двоичные данные без знака между 0000 и FFFF (шестн.), не позволяя непосредственно использовать команду SCL(194) для обработки двоичных данных, выражающих значение ниже 1В (0000 шестн.), т.е. FF38...FFFF (шестн.). Поэтому в реальной ситуации необходимо прибавить значение 00C8 (шестн.) ко всем значениям, так, чтобы перед использованием команды SCL(194) значение FF38 было представлено значением 0000 (шестн.), как показано в следующем примере.



Содержание D00200 R

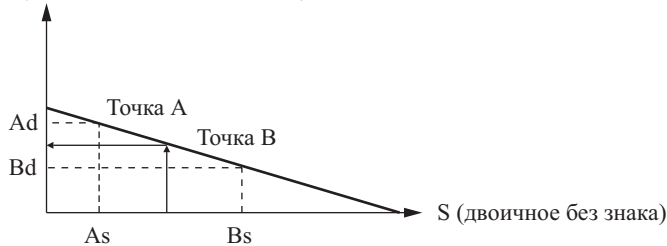


В данном примере значения от 0000 до 00C8 (шестн.) будут преобразованы в отрицательные значения. Тем не менее, команда SCL(194) может выводить только двоично-десятичные числа без знака от 0000 до 9999, поэтому двоично-десятичное число 0000 будет выводиться в любом случае, когда содержание D00000 будет находиться между значениями от 0000 до 00C8 (шестн.)

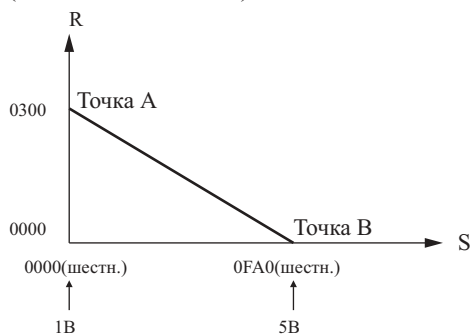
**Обратное масштабирование**

Обратное масштабирование может вводиться посредством задания точек, подчиняющихся соотношениям  $A_s < B_s$  и  $A_r > B_r$ . При этом линейной характер зависимости приобретает следующий вид.

R (двоично-десятичное без знака)



Обратное масштабирование может использоваться, например, для преобразования значений от 1 до 5 В (0000...0FA0 шестн.) в значения от 0300 до 0000 соответственно, как показано на следующем рисунке.



## 3-17-6 Команда преобразования данных SCALING: SLC2(486)

**Назначение**

По команде SLC2(486) производится преобразование двоичных данных со знаком в двоично-десятичные данные со знаком, соответственно указанной линейной зависимости. При задании линейной зависимости может вводиться смещение.

**Символ релейно-контактной схемы**

SCL2(486)	
S	S: Исходное слово
P1	P1: Слово первого параметра
R	R: Результирующее слово

**Модификации**

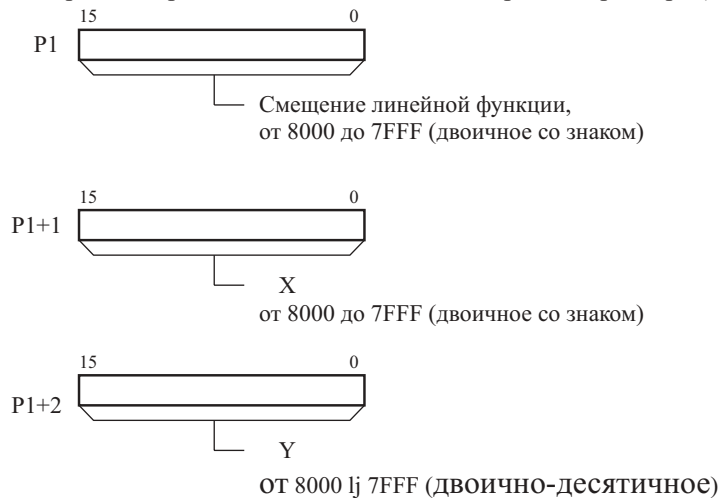
Модификации	Выполняется в каждом цикле для состояния ON	SLC2(486)
	Выполняется один раз для дифференцирования вверх	@ SLC2(486)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды**

Содержание трех слов, начиная со слова первого параметра (P1), показано на следующем ниже рисунке.

**Спецификации операндов**

Область	S	C	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6141	CIO 0000...CIO 6143
Рабочая область	W000...W511	W000...W509	W000...W511
Область удержания бита (Holding bit area)	H000...H511	H000...H509	H000...H511
Область вспомогательных битов	A000...A959	A000...A957	A448...A959
Область таймера	T0000...T4095	T0000...T4093	T0000...T4095
Область счетчика	C0000...C4095	C0000...C4093	C0000...C4095
Область DM	D00000...D32767	D00000...D32765	D00000...D32767
Область EM, не содержащая банков	E00000...E32767	E00000...E32765	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32765 (n = 0...C)	En_00000...En_32767 (n = 0...C)



Область	S	C	D
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	–		
Регистры данных	DR0...DR15	–	DR0...DR15
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-)IR15		

### Описание

Команда SCL2(486) применяется для преобразования двоичных данных со знаком, содержащихся в исходном слове S в двоично-десятичные данные со знаком, и вывода результата в слово R согласно линейной зависимости, определяемой отношением  $\Delta X$ ,  $\Delta Y$  и смещением. Адрес слов, содержащих  $\Delta X$  и  $\Delta Y$ , а также смещение, указывается Словом первого параметра P1. Знак результата обозначается состоянием флага переноса (ON: отрицательный, OFF: положительный.)

Для преобразования используются следующие ниже уравнения.

$$R = \frac{\Delta Y}{\Delta X} \times (S - Os)$$

S и Os (смещение) в двоично-десятичном коде.

Крутизна (наклон) характеристики равна

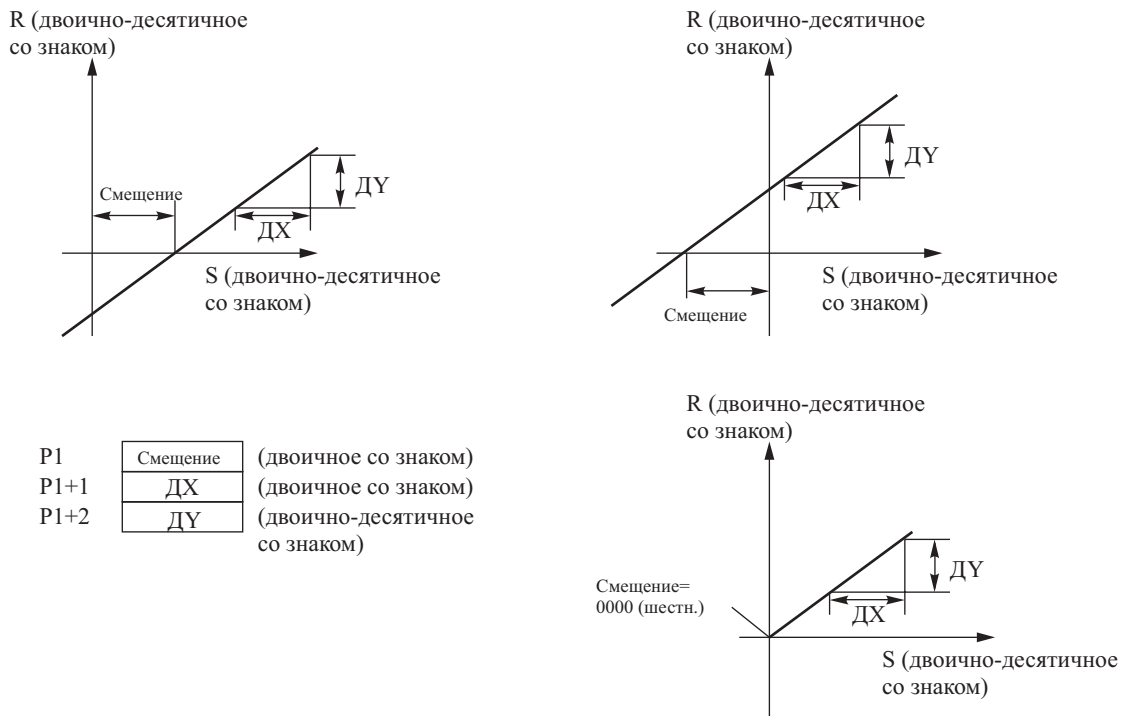
$$K = \frac{\Delta Y}{\Delta X}$$

Смещение и наклон характеристики линейной зависимости могут иметь положительное значение, отрицательное значение или могут быть равны нулю. Отрицательное значение наклона характеристики обеспечивает обратное масштабирование.

Результат округляется до ближайшего целого числа.

Результат в слове R будет представлен абсолютным значением в двоично-десятичном коде, а знак определяется состоянием флага переноса. Таким образом, результат будет находиться между значениями от -9999 до +9999.

Если результат меньше -9999, в качестве результата выводится значение, равное -9999. Если результат больше 9999, в качестве результата выводится значение, равное 9999.



Команда SCL2(486) может использоваться для преобразования выходных сигналов Модулей аналогового ввода в величину, определяемую пользователем. Например, если входные напряжения от 1 до 5В, подаваемые на вход Модуля аналогового ввода, вводятся в память как 0000...0FA0 (шестн.), при помощи команды SCL2(486) эти значения могут быть преобразованы в значения от -100 до +200С.

По команде SCL2(486) производится преобразование двоичных данных со знаком в двоично-десятичные данные со знаком, поэтому отрицательные значения S могут обрабатываться непосредственно. Команда SCL2(486) также может выводить в результирующее слово отрицательные значения.

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержанием C+1 ( $\Delta X$ ) является 0000. Переводится в состояние ON, когда содержание C+2 ( $\Delta Y$ ) выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, если результат равен нулю. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда результат является отрицательным числом. OFF в других случаях.

#### Меры предосторожности

В случае, когда значение  $\Delta X$  (C+1) равно 0000 или значение  $\Delta Y$  (C+2) выражены не в двоично-десятичном коде, определяется ошибка и флаг ошибки переводится в состояние ON.

Если содержанием результирующего слова D равно 0000, флаг равенства переводится в состояние ON.

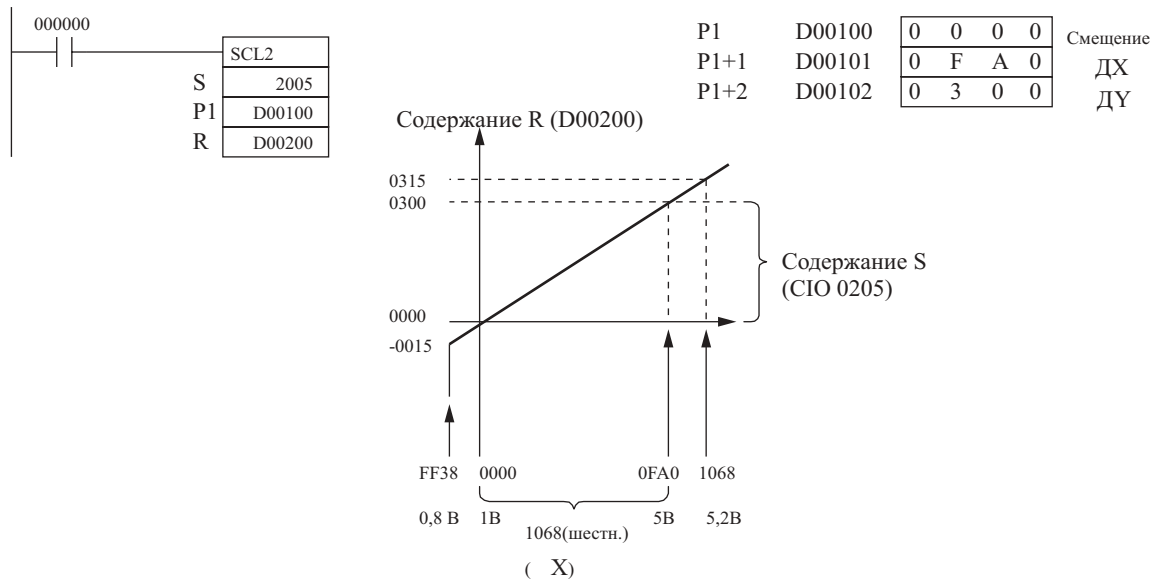
Если содержанием результирующего слова является отрицательное число (т.е. когда бит старшего разряда = 1), флаг переноса переводится в состояние ON.

#### Примеры

##### Масштабирование входного сигнала Модуля аналогового ввода от 1 до 5В в значения от 0 до 300

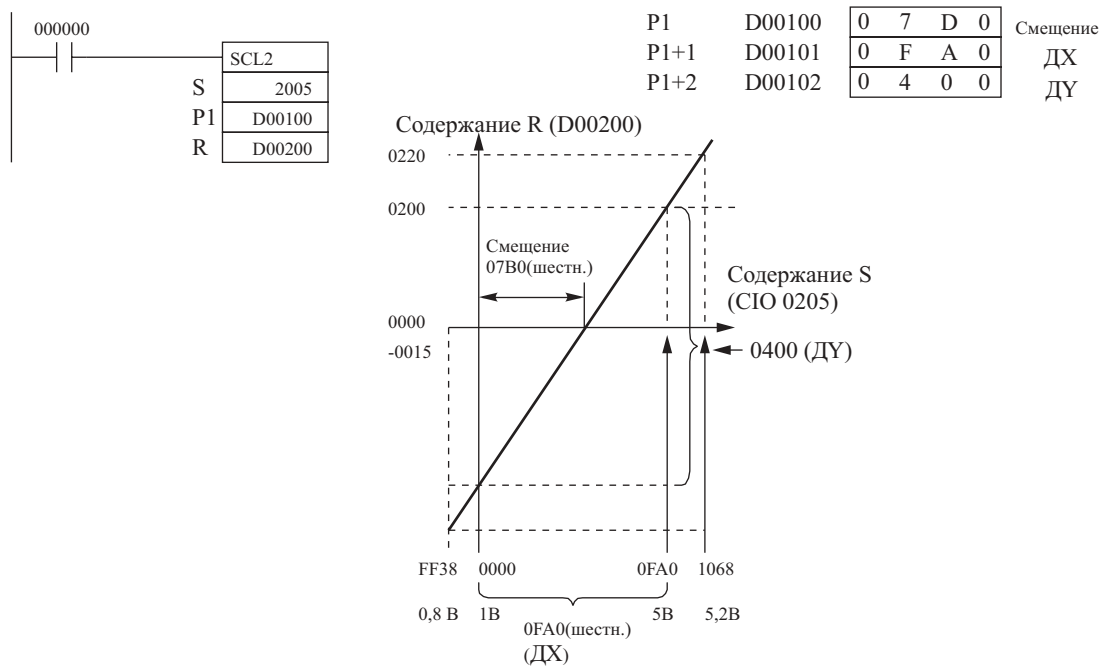
В следующем ниже примере предполагается, что аналоговый сигнал размахом от 1 до 5 В преобразуется и заносится в СЮ 0205 в виде значений от 0000 до 0FA0 (шестн.) Команда SCL2(486) используется для преобразования (масштабирования) содержания СЮ 0205 в значения от 0000 до 0300 в двоично-десятичном коде.

Когда СЮ 000000 находится в состоянии ON, содержание СЮ 0205 масштабируется с использованием линейной зависимости, определяемой  $\Delta X$  (0FA0),  $\Delta Y$  (0300), и смещением (0). Эти значения содержатся в D00100...D00102. Результат выводится в D00200.



**Масштабирование входного сигнала Модуля аналогового ввода от 1 до 5В в значения от -200 до 300**  
 В следующем ниже примере предполагается, что аналоговый сигнал размахом от 1 до 5 В преобразуется и заносится в CIO 0205 в виде значений от 0000 до 0FA0 (шестн.) Команда SCL2(486) используется для преобразования (масштабирования) содержания CIO 0205 в значения от -0200 до 0200 в двоично-десятичном коде.

Когда CIO 000000 находится в состоянии ON, содержание CIO 0205 масштабируется с использованием линейной зависимости, определяемой ΔX (0FA0), ΔY (0400), и смещением (07D0). Эти значения содержатся в D00100...D00102. Результат выводится в D00200.



### 3-17-7 Команда преобразования данных SCALING: SLC3(487)

#### Назначение

По команде SLC3(487) производится преобразование двоично-десятичных данных со знаком в двоичные данные со знаком, соответственно указанной линейной зависимости. При задании линейной зависимости может вводиться смещение.

**Символ релейно-контактной схемы**

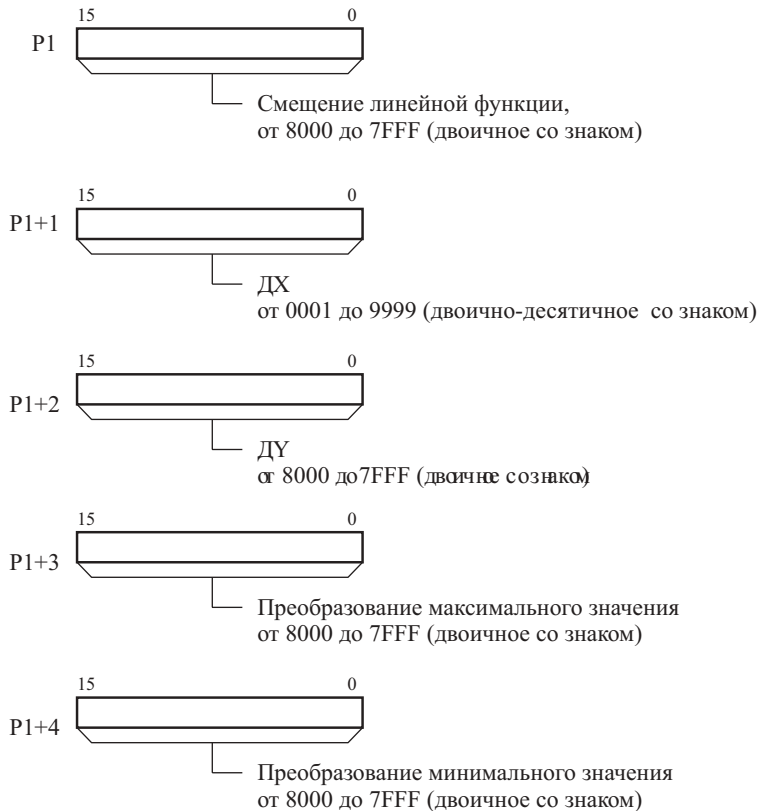
SCL3(487)	
S	S: Исходное слово
P1	P1: Слово первого параметра
R	R: Результирующее слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	SLC3(487)
	Выполняется один раз для дифференцирования вверх	@ SLC3(487)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды**

**Примечание:** P1...P1+4 должны находиться в одной области данных.

**Спецификации операндов**

Область	S	C	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6139	CIO 0000...CIO 6143
Рабочая область	W000...W511	W000...W507	W000...W511
Область удержания бита (Holding bit area)	H000...H511	H000...H507	H000...H511
Область вспомогательных битов	A000...A447 A448...A959	A000...A443 A448...A955	A448...A959
Область таймера	T0000...T4095	T0000...T4091	T0000...T4095
Область счетчика	C0000...C4095	C0000...C4091	C0000...C4095

Область	S	C	D
Область DM	D00000...D32767	D00000...D32763	D00000...D32767
Область EM, не содержащая банков	E00000...E32767	E00000...E32763	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32763 (n = 0...C)	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	–		
Регистры данных	DR0...DR15	–	DR0...DR15
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-( -)IR0...,-( -)IR15		

### Описание

Команда SCL3(487) применяется для преобразования двоично-десятичных данных со знаком, содержащихся в исходном слове S, в двоичные данные со знаком, и вывода результата в слово R согласно линейной зависимости, определяемой отношением  $\Delta X$ ,  $\Delta Y$  и смещением. Указываются также максимальное и минимальное значения преобразования. Адрес слов, содержащих  $\Delta X$  и  $\Delta Y$ , смещение, максимальное значение преобразования и минимальное значение преобразования указывается словом первого параметра P1. Знак результата обозначается состоянием флага переноса (ON: отрицательный, OFF: положительный.) Для преобразования используются следующие ниже уравнения.

$$R = \frac{\Delta Y}{\Delta X} \times (S + Os)$$

$\Delta X$  и S в двоичном коде.

Крутизна (наклон) характеристики равна

$$K = \frac{\Delta Y}{\Delta X}$$

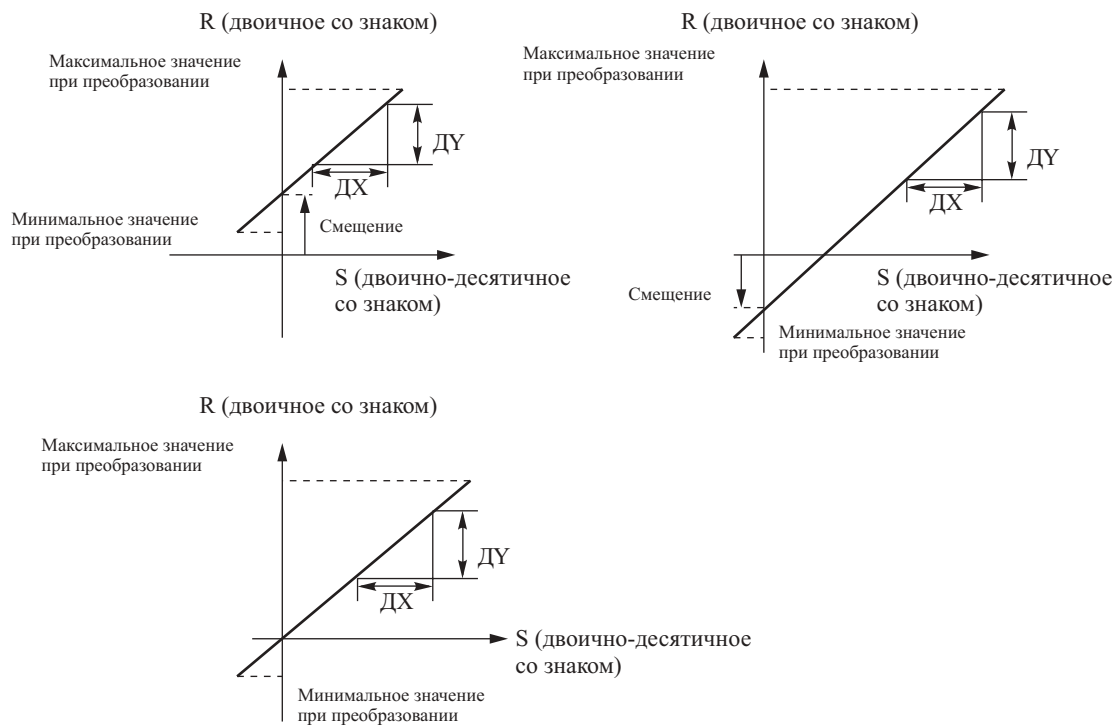
Смещение и наклон характеристики линейной зависимости могут иметь положительное значение, отрицательное значение или могут быть равны нулю. Отрицательное значение наклона характеристики обеспечивает обратное масштабирование.

Результат округляется до ближайшего целого числа.

Исходное слово S обрабатывается как абсолютное значение в двоично-десятичном коде, знак числа указывается флагом переноса.

Таким образом, исходное число будет находиться между значениями от –9999 до +9999.

Если результат меньше минимального значения при преобразовании, в качестве результата выводится минимальное значение. Если результат больше максимального значения при преобразовании, в качестве результата выводится максимальное значение.



Команда SCL3(487) применяется для преобразования данных с использованием масштабирования, определяемого пользователем, в двоичные данные со знаком для Модулей аналогового вывода. Например, при помощи команды SCL3(487) можно преобразовать значения температуры от 0 до +200С в значения от 0000 до 0FA0 (шестн.) и вывести аналоговый выходной сигнал от 1 до 5В на выход Модуля аналогового вывода.

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание S выражено не в двоично-десятичном коде. Переводится в состояние ON, когда содержание S+1 (Δ X) находится за пределами значений от 0001 до 9999 (в двоично-десятичном коде). OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, если результат равен нулю. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда результат является отрицательным числом (т.е. старший бит слова R равен 1). OFF в других случаях.

#### Меры предосторожности

В случае, когда содержание S выражено не в двоично-десятичном коде или когда содержание S+1 (Δ X) находится за пределами значений от 0001 до 9999 (в двоично-десятичном коде), определяется ошибка и флаг ошибки переводится в состояние ON.

Если содержанием результирующего слова D равно 0000, флаг равенства переводится в состояние ON.

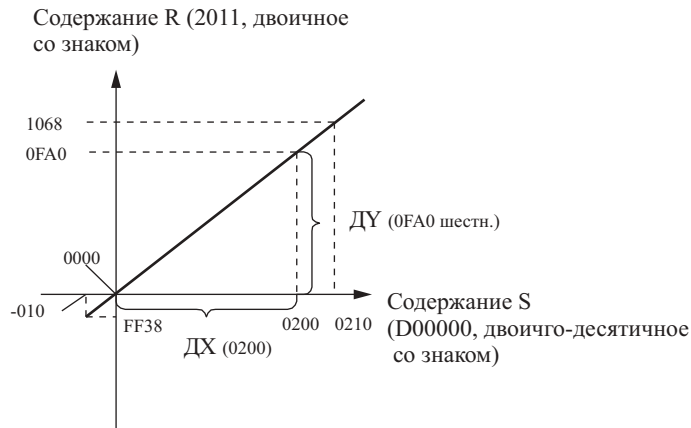
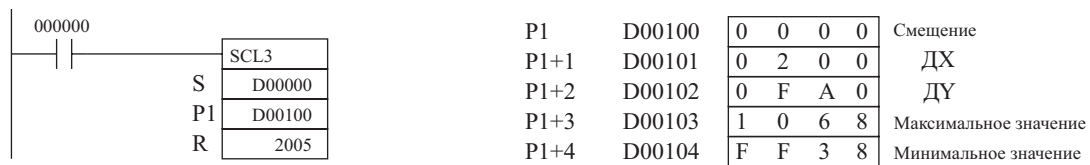
Если содержанием результирующего слова является отрицательное число (т.е. когда бит старшего разряда = 1), флаг переноса переводится в состояние ON.

#### Примеры

Когда число в диапазоне значений от 0 до 200 масштабируется в аналоговый сигнал (например, от 1 до 5В), двоично-десятичное число со знаком (от 0000 до 0200) преобразуется в двоичное число со знаком (от 0000 до 0FA0) для Модуля аналогового вывода.

В следующем ниже примере, когда CIO 000000 переводится в состоянии ON, содержание D00000 масштабируется с использованием линейной зависимости, определяемой ΔX (0200), ΔY (0FA0), и смещением (0).

Эти значения содержатся в D00100...D00102. Знак двоично-десятичного числа указывается состоянием флага переноса. Результат выводится в D00200.

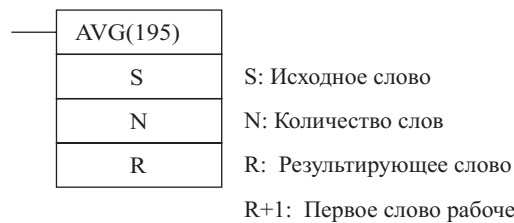


### 3-17-8 Команда вычисления среднего значения AVERAGE: AVG(195)

#### Назначение

По команде AVG(195) производится вычисление среднего значения входного слова в указанном количестве циклов.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	AVG(195)
	Выполняется один раз для дифференцирования вверх	Не поддерживается
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не используется	Да	Да	Да

#### Операнды

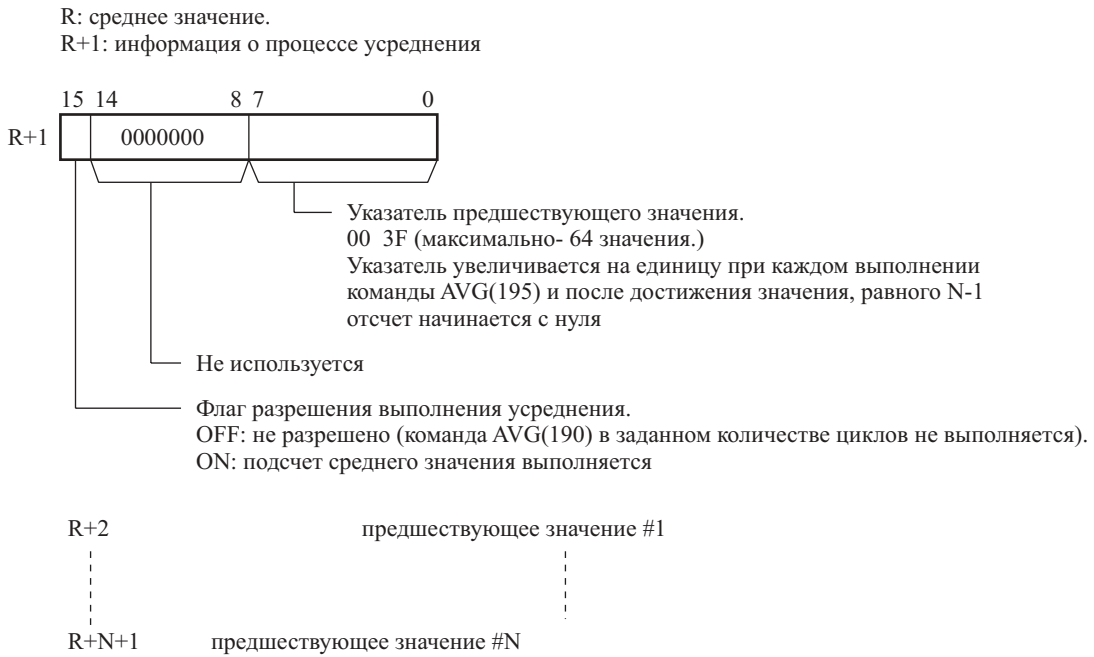
##### N: Количество циклов

Количество циклов должно находиться между 0001 и 0040 (шестн.) (от 0 до 64 циклов).

##### R: Результирующее слово и R+1: Первое слово рабочей области

После истечения заданного количества циклов слово R содержит среднее значение.

Слово R1 содержит информацию о процессе усреднения, а слова R+2...R+N+1 содержат предшествующие значения, как показано на следующем ниже рисунке.



**Примечание:** R1...R+N+1 должны находиться в одной области данных.

### Спецификации операций

Область	S	N	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A447 A448...A959		A448...A959
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF (двоичные)	#0001...#0040 (двоичные)	–
Регистры данных	DR0...DR15	–	
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15		

### Описание

Для первых N-1 циклов, когда условие выполнения команды находится в состоянии ON, по команде AVG(195) значение S записывается в слова, начиная с R+2 и далее по порядку. Указатель предшествующего значения (биты 00...07 слова R+1) увеличивается каждый раз при записи очередного значения. До тех пор пока не будет записано значение номер N, содержание S остается в слове R неизменным, а флаг разрешения вычисления среднего значения (бит 15 слова R+1) остается в состоянии OFF. Когда значение номер N записывается в R+N+1, производится вычисление среднего значения всех записанных слов, которое записывается в R в виде двоичного числа без знака. При этом флаг среднего значе-

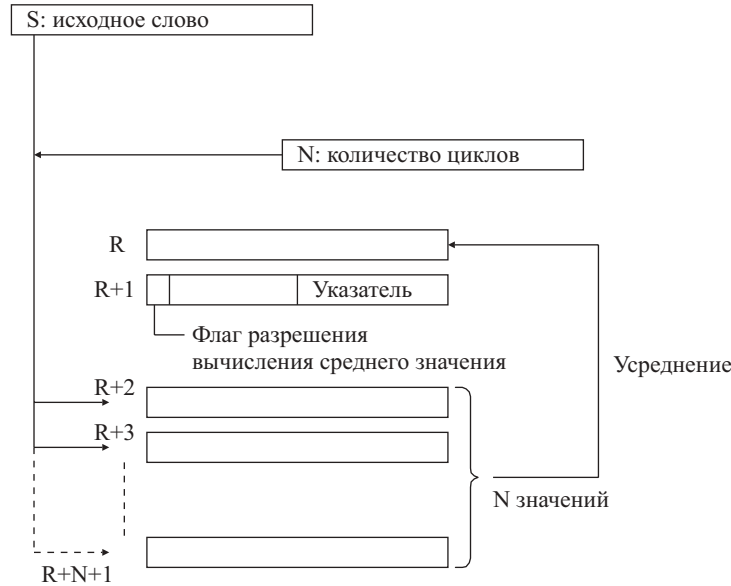


ния (бит 15 слова R+1) переводится в состояние ON. Для всех последующих циклов значение, находящееся в R, обновляется для N текущих значений S.

Максимальное значение N = 64.

Указатель предшествующего значения сбрасывается в ноль после записи N-1 значений.

Среднее значение, записываемое в R, округляется до ближайшего целого числа.



#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда содержание N равно нулю. OFF в других случаях.

#### Меры предосторожности

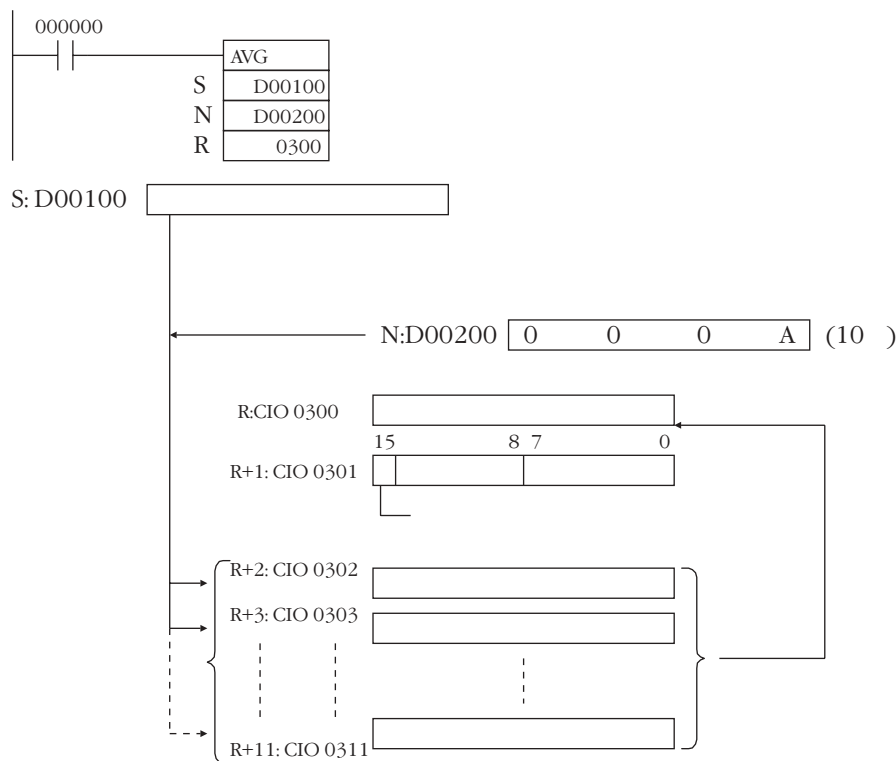
Содержание первого слова рабочей области (D+1) обнуляется каждый раз, когда условие выполнения переходит из состояния OFF в состояние ON.

Содержание первого слова рабочей области в начале выполнения программы не очищается. Если команда AVG(195) должна выполняться в первом сканировании, выполняйте обнуление первого слова рабочей области из программы.

Если N (количество циклов) содержит значение 0000, определяется ошибка и флаг ошибки переводится в состояние ON.

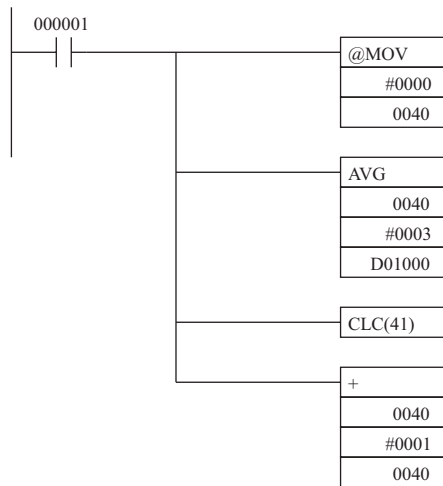
В следующем ниже примере, когда CIO 000000 переводится в состояние ON, содержание слова D00100 будет записано один раз при каждом сканировании. Количество сканирований указывается в D00200. Содержание слов записывается по порядку в десяти словах от CIO 0302 до CIO 0311.

Среднее значение этих десяти слов располагается в CIO 0300 и затем бит 15 слова CIO 0301 переводится в состояние ON.



**Примеры**

В следующем ниже примере, содержание CIO 00400 установлено в значение # 0000 и увеличивается на единицу в каждом цикле. При выполнении первых двух циклов команда AVG(195) перемещает содержание CIO 0040 в D01002 и D01003. Содержание D01001 будет также изменено, (что может использоваться для подтверждения того, что после выполнения команды AVG(195) результат изменился). При выполнении третьего цикла и в последующих циклах, по команде AVG(195) производится вычисление среднего значения содержания D01002...D01004 и запись среднего значения в D01000.



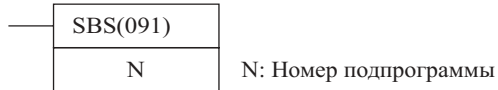
D01000	0000	0001	0001	0002	Указатель среднего значения
D01001	0001	0002	8000	8001	
D01002	0000	0000	0000	0003	Три предшествующих значения IR40
D01003	—	0001	0001	0001	
D01004	—	—	0002	0002	

## 3-18 Подпрограммы

## 3-18-1 Команда вызова подпрограммы SUBROUTINE CALL: SBS(091)

**Назначение**

По команде SBS(091) производится вызов и выполнение подпрограммы с указанным номером.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	SBS(091)
	Выполняется один раз для дифференцирования вверх	@ SBS(091)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****N: Номер подпрограммы**

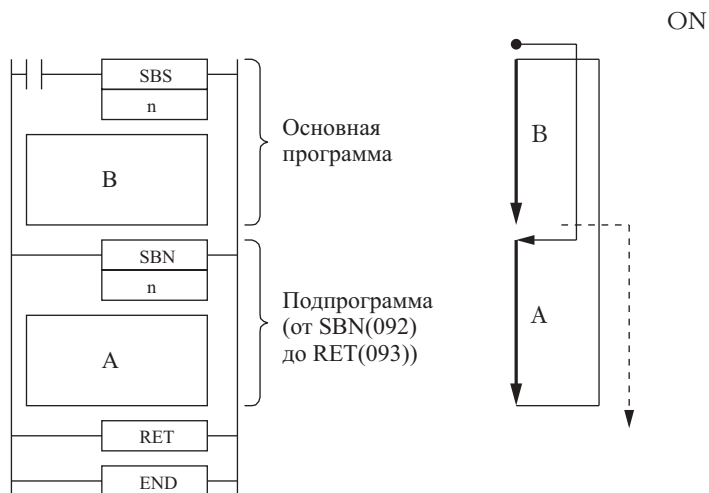
Указывает номер подпрограммы от 0 до 1023 в десятичном коде.

**Спецификации операндов**

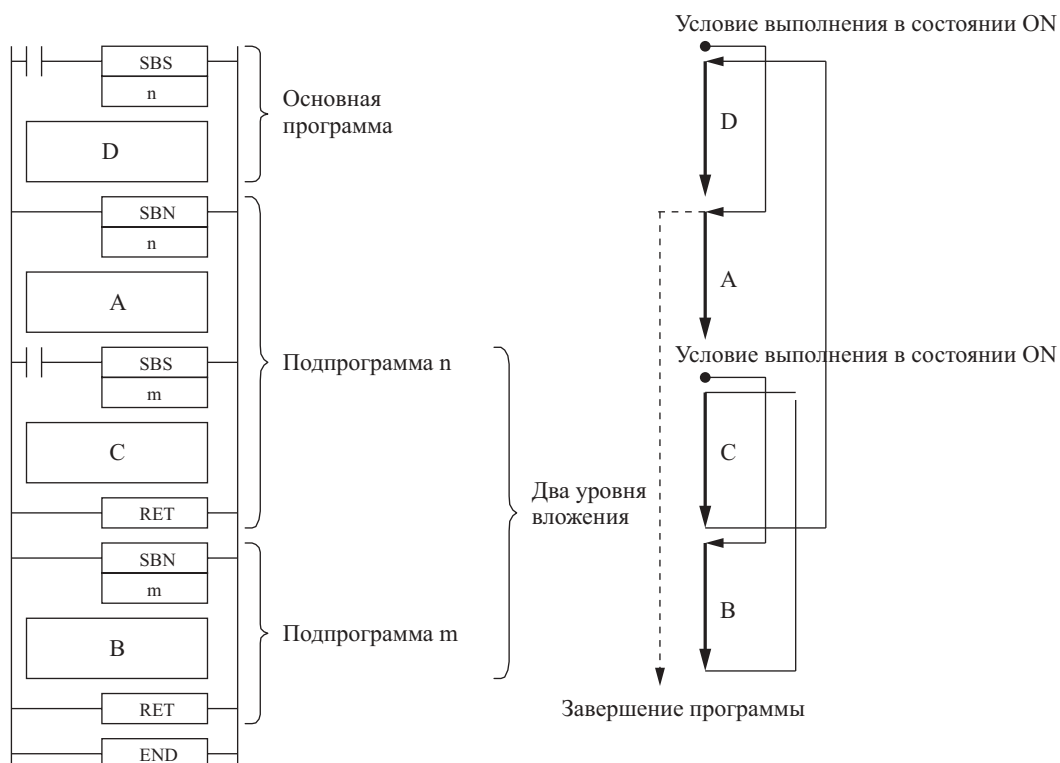
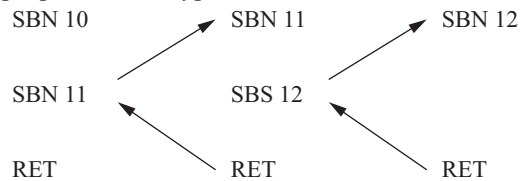
Область	N
Область ввода/вывода (область CIO)	—
Рабочая область	—
Область удержания бита (Holding bit area)	—
Область вспомогательных битов	—
Область таймера	—
Область счетчика	—
Область DM	—
Область EM, не содержащая банков	—
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	От 0 до 1023
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	—

**Описание**

По команде SBS(091) производится вызов подпрограммы с указанным номером. Подпрограмма - это раздел программы между командами SBN(092) и RET(093). Когда выполнение подпрограммы завершается, выполнение программы возобновляется с команды, следующей за SBS(091).



Подпрограммы могут вкладываться одна в другую вплоть до 16 уровней. Вложение – это ситуация, когда одна подпрограмма вызывается из другой подпрограммы, например, как показано на следующем рисунке, где производится вложение подпрограмм до 3-х уровней.

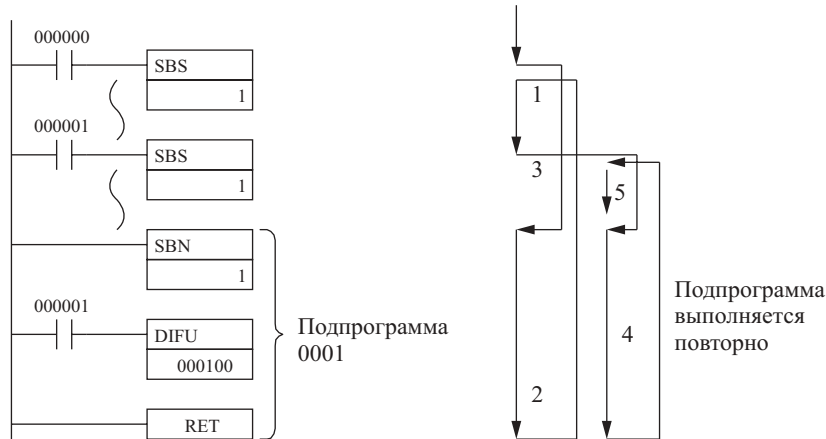


**Примечание:** Подпрограмма может вызываться из программы несколько раз.

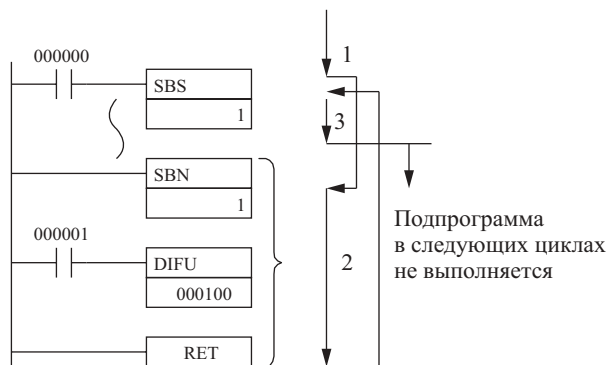
#### Подпрограммы и дифференцированные команды

При использовании в подпрограммах команд с дифференцированием (DIFU(013), DIFU(014), а также команд дифференцированных вниз/вверх) изучите следующие меры предосторожности.

Поведение дифференцированных команд в подпрограммах непредсказуемо, если эта подпрограмма выполняется в одном цикле более одного раза. В следующем ниже примере подпрограмма 0001 выполняется, когда CIO 000000 находится в состоянии ON, а CIO 000100 переводится в состояние ON по команде DIFU(013) когда CIO 000001 переходит из состояния OFF в состояние ON. Если CIO 000001 находится в состоянии ON в этом же цикле, подпрограмма 0001 будет выполняться повторно, однако на этот раз команда DIFU(013) переводит CIO 000100 в состояние OFF без проверки состояния CIO 000001.



В противоположность показанному примеру, выход дифференцированной команды (DIFU(013) и DIFU(014)) будет оставаться в состоянии ON, если команда выполнена и вывод переведен в состояние ON, однако та же подпрограмма не вызвана повторно.



В следующем примере подпрограмма 0001 выполняется, если CIO 000000 находится в состоянии ON. Выход CIO 000100 переводится в состояние ON командой DIFU(013) когда CIO 000001 переходит из состояния OFF в состояние ON. Если в следующем цикле CIO 000000 находится в состоянии OFF, подпрограмма 0001 выполняться не будет и вывод CIO 000100 останется в состоянии ON.

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	<p>Переводится в состояние ON, когда вложения превышают 16 уровней.</p> <p>Переводится в состояние ON, когда вызываемая подпрограмма не существует.</p> <p>Переводится в состояние ON, когда подпрограмма вызывает саму себя.</p> <p>Переводится в состояние ON, когда вызывается подпрограмма, которая выполняется.</p> <p>Переводится в состояние ON, когда указанная подпрограмма в данной задаче не определена.</p> <p>OFF в других случаях.</p>

**Примечание:** Метод ввода номера подпрограммы различен для СХ- программатора и Пульта программирования. С помощью СХ- программатора вводите номера от 0 до 1023 или от &0 до &127. С помощью пульта программирования вводите номера от 0000 до 1023.

**Меры предосторожности**

Команда SBS(091) и соответствующая ей команда SBN(092) должны программироваться в одной задаче. Если в задаче отсутствует соответствующая команда SBN(092) определяется ошибка.

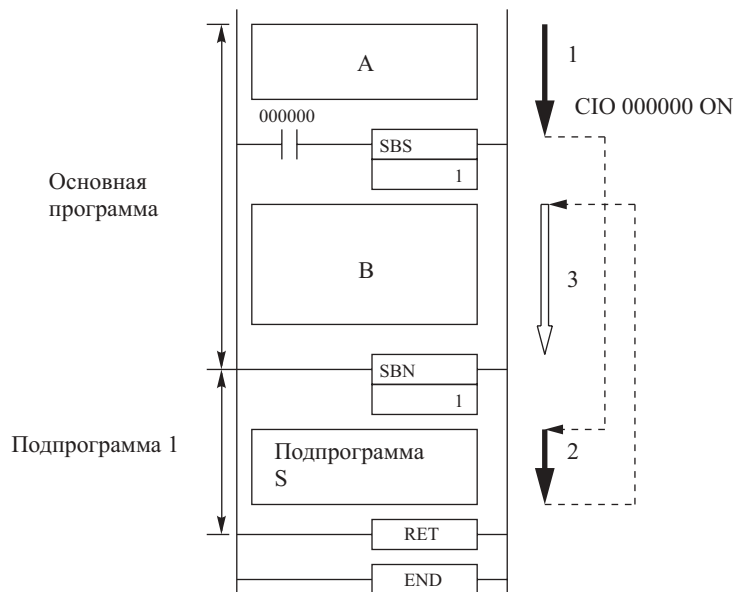
Команда SBS(091) выполняется как команда NOP(000), если она находится в части программы, блокированной командами IL(002) и ILC(003).

В следующих ниже случаях подпрограмма вызываться не будет, даже если выполняется команда SBS(091).

- 1,2,3... 1. Указываемая подпрограмма не определена в текущей задаче.
2. Подпрограмма вызывает саму себя.
3. Вложение подпрограмм превышает 16 уровней.
4. Указанная подпрограмма выполняется.

**Примеры****Пример 1: Последовательные (не вложенные) подпрограммы**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, выполняется подпрограмма 1 и затем программа переходит к выполнению следующей команды, находящейся после команды SBS(091). Затем выполняется оставшаяся часть основной программы (до команды, находящейся перед SBN(092)).

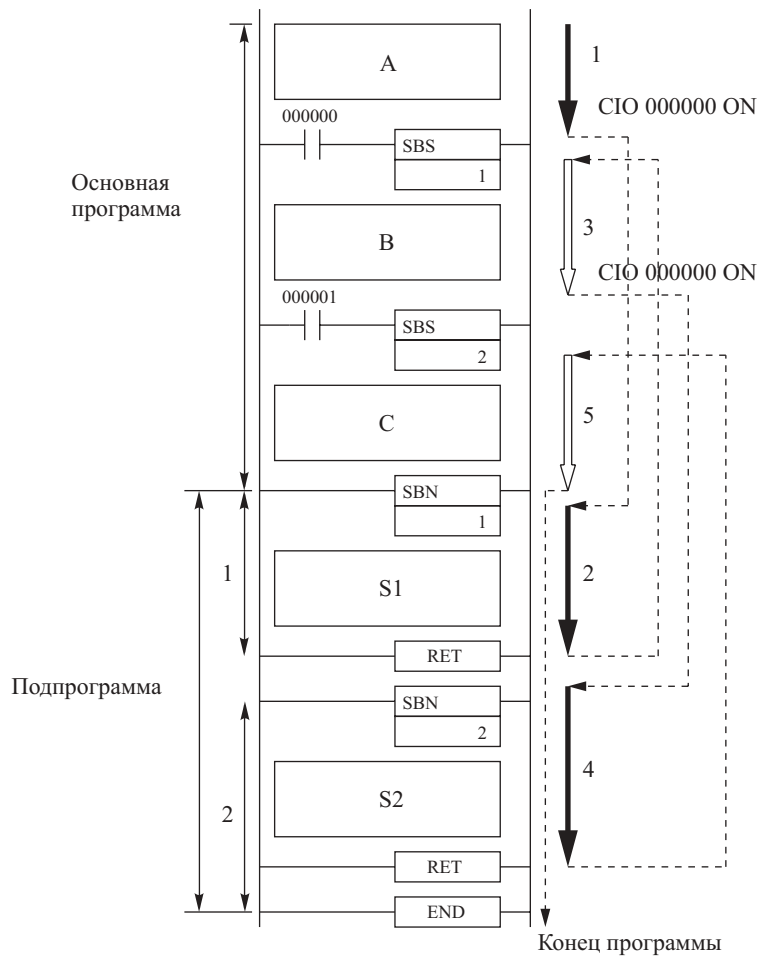


000000	Порядок выполнения
ON	A→S→B
OFF	A→B

**Пример 2: Последовательные (не вложенные) подпрограммы**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, выполняется подпрограмма 1 и затем программа переходит к выполнению команды, находящейся после команды SBS(091) 1.

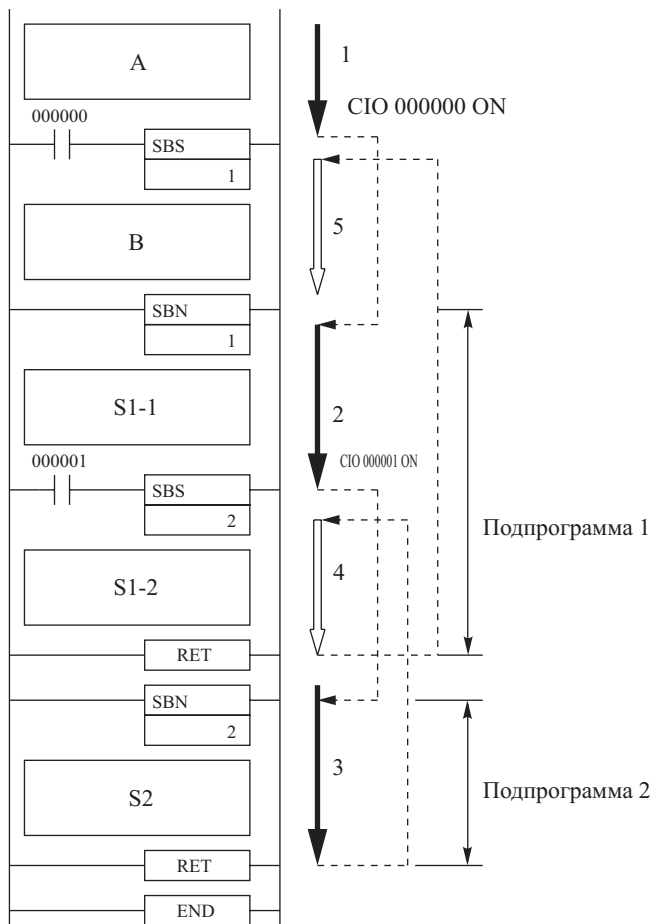
Когда CIO 000001 находится в состоянии ON, выполняется подпрограмма 2 и программа переходит к выполнению следующей команды, находящейся после SBS(091) 2 команды



000000	000001	Порядок выполнения
ON	ON	AS1BS2C
ON	ON	A→S1→B→S2→C
ON	OFF	A→S1→B→C
OFF	ON	A→B→S2→C
OFF	OFF	A→B→C

### Пример 3: Вложенные подпрограммы

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, выполняется подпрограмма 1. Если CIO000001 находится в состоянии ON, подпрограмма 2 выполняется из подпрограммы 1 и после завершения подпрограммы 2 программа переходит к выполнению команды, следующей после команды SBS(091) 2. Выполнение подпрограммы 1 продолжается и после ее завершения программа переходит к выполнению следующей за SBS(091)1 команды.



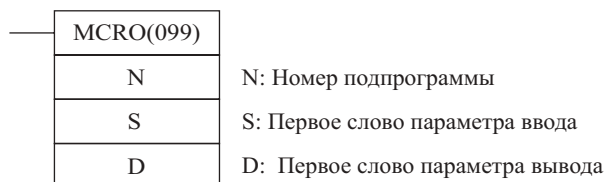
000000	000001	Порядок выполнения
ON	ON	A→S1-1→S2→S1-2→B
ON	OFF	A→S1-1→S1-2→B
OFF	ON	A→B
OFF	OFF	A→B

### 3-18-2 Команда MACRO: MCRO(099)

#### Назначение

По команде MCRO(099) осуществляется вызов подпрограммы с заданным номером и выполнение этой подпрограммы, используя параметры ввода в словах S S+3 и параметры вывода в D D+3.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	MCRO(099)
	Выполняется один раз для дифференцирования вверх	@ MCRO(099)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	



**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****N: Номер подпрограммы**

Указывает номер подпрограммы от 0 до 1023 в десятичном коде.

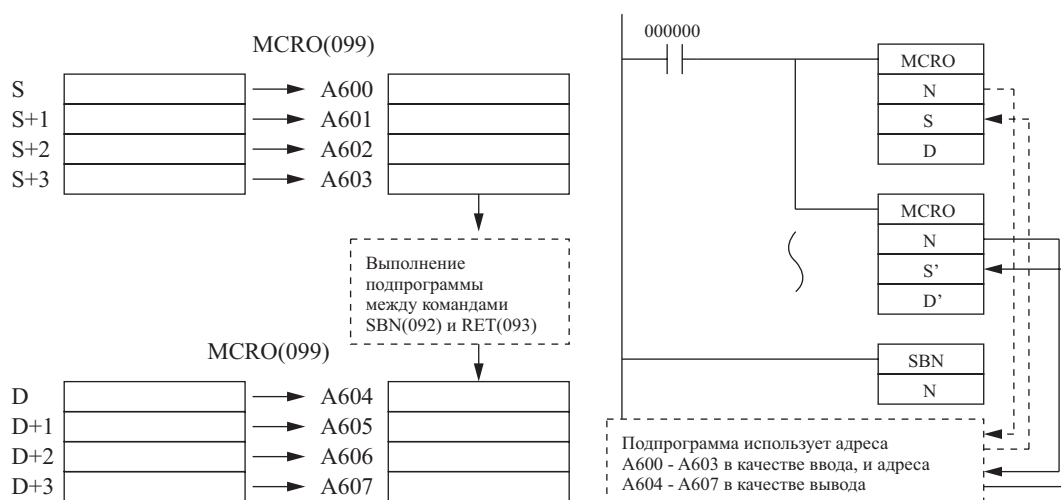
**Спецификации операндов**

Область	N	S	D
Область ввода/вывода (область CIO)	–	CIO 0000...CIO 6140	
Рабочая область	–	W000...W508	
Область удержания бита (Holding bit area)	–	H000...H508	
Область вспомогательных битов	–	A000...A444 A448...A956	A448...A956
Область таймера	–	T0000...T4092	
Область счетчика	–	C0000...C4092	
Область DM	–	D00000...D32764	
Область EM, не содержащая банков	–	E00000...E32764	
Область EM, содержащая банки	–	En_00000...En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	–	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	От 0 до 1023	–	
Регистры данных	–		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров		,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(-)IR0...,-(-)IR15	

**Описание**

По команде MCRO(099) производится вызов подпрограммы, имеющей указанный номер, подобно тому, как это производится по команде SBS(091). В отличие от команды SBS(091), операнды S и D могут использоваться в подпрограмме для изменения адресов битов и слов, хотя при этом структура подпрограммы остается неизменной.

При выполнении команды MCRO(099) содержание слов S...S+3 копируется в A600...A603 (входы области макро) и указанная подпрограмма выполняется. По завершении выполнения подпрограммы, содержание A604...A607 (выводы области макро) копируются в D...D+3 и программа переходит к выполнению команды, следующей за MCRO(099).



Команда MCRO(099) может применяться для объединения в единую подпрограмму двух и более подпрограмм, имеющих одинаковую структуру, однако различные адреса ввода и вывода. При выполнении команды MCRO(099) заданные входные и выходные данные передаются в указанную подпрограмму.

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	<p>Переводится в состояние ON, когда вложения превышают 16 уровней.</p> <p>Переводится в состояние ON, когда вызываемая подпрограмма не существует.</p> <p>Переводится в состояние ON, когда подпрограмма вызывает сама себя.</p> <p>Переводится в состояние ON, когда вызывается подпрограмма, которая выполняется.</p> <p>Переводится в состояние ON, когда указанная подпрограмма в данной задаче не определена.</p> <p>OFF в других случаях.</p>

В следующей ниже таблице приводятся слова вспомогательной области, относящиеся к данной команде.

Наименование	Адрес	Функционирование
Слова ввода области макроса	A600...A603	При выполнении команды MCRO(099) четыре слова от S до S+3 записываются в A600...A603. Эти слова ввода передаются подпрограмме.
Слова вывода области макроса	A604...A607	После выполнения подпрограммы, указанной в MCRO(099) выходные данные в этих словах записываются в D...D+3.

### Меры предосторожности

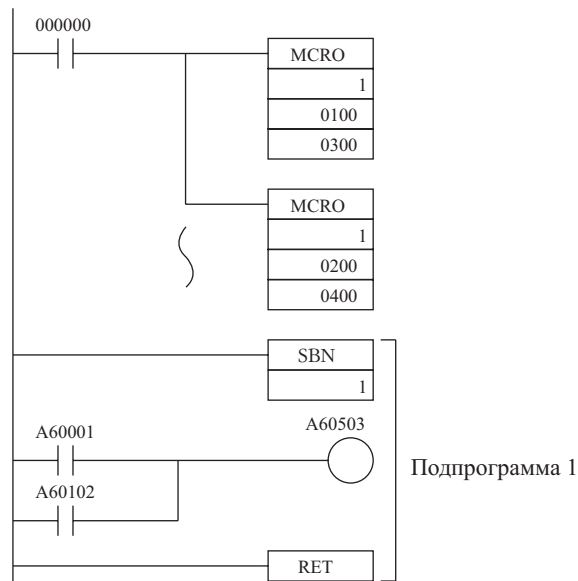
Четыре слова данных ввода (слова или биты) в A600...A603 и четыре слова данных вывода (слова или биты) в A604...A607 должны использоваться в подпрограмме, которую вызывает команда MCRO(099). Передача большего количества слов невозможна.

Существует возможность вложения команд MCRO(099), однако данные в области макроса, т.е. входные и выходные слова A600...A607, должны сохраняться до вызова следующей подпрограммы, так как все команды MCRO(099) используют одни и те же 8 слов.

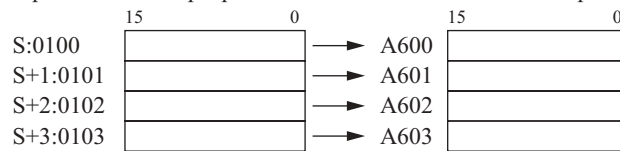
### Пример

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, две команды MCRO(099) передают различные данные ввода и вывода подпрограмме 1.

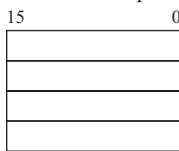
- 1,2,3...**
1. Первая из двух команд MCRO(099) передает данные ввода в CIO 0100...CIO 0103 и выполняет подпрограмму. После завершения выполнения подпрограммы, данные вывода записываются в CIO 0300...CIO 0303.
  2. Вторая команда MCRO(099) передает данные ввода в CIO 0200...CIO 0203 и выполняет подпрограмму. После завершения выполнения подпрограммы данные вывода записываются в CIO 0400...CIO 0403.



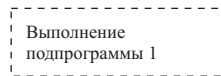
Данные ввода передаются при вызове подпрограммы



Слова ввода области макроса

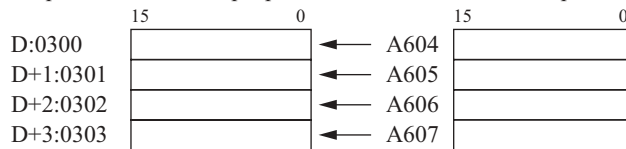


↓ Ввод

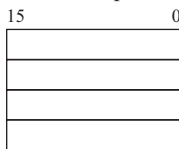


↓ Вывод

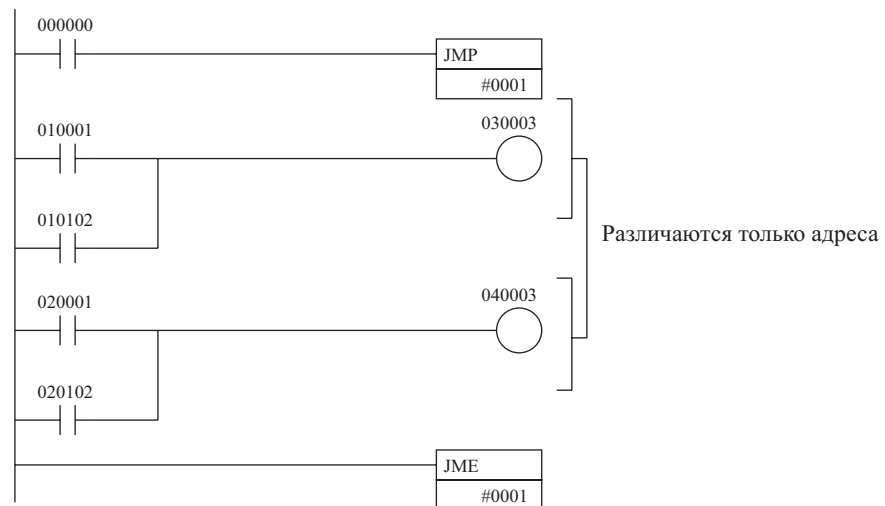
Данные вывода передаются при выходе из подпрограммы.



Слова вывода области макроса



Вторая команда MCRO(099) функционирует аналогичным образом, однако входные данные из CIO 0200...CIO 0203 передаются в A600...A603 и выходные данные из A604 A607 передаются в CIO 0400 CIO 0403.

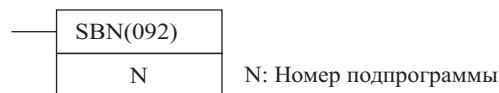


### 3-18-3 Команда начала выполнения подпрограммы SUBROUTINE ENTRY: SBN(092)

#### Назначение

Команда SBN(092) обозначает начало выполнения подпрограммы с указанным номером. Данная команда применяется совместно с командой RET(093) для определения области подпрограммы.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	SBN(092)
	Выполняется один раз для дифференцирования вверх	@ SBN(092)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не используется	Не используется	См. примечание	Да

**Примечание:** Допускается использование только одной команды SBN(092) в начале каждой из подпрограмм.

#### Операнды

##### N: Номер подпрограммы

Указывает номер подпрограммы от 0 до 1023 в десятичном коде.

#### Спецификации операндов

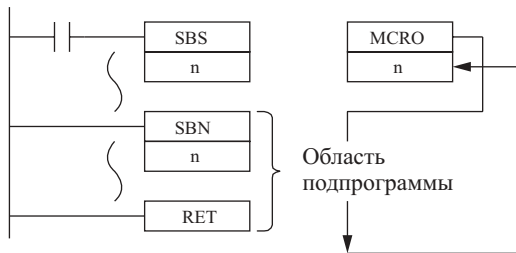
Область	N
Область ввода/вывода (область CIO)	—
Рабочая область	—
Область удержания бита (Holding bit area)	—
Область вспомогательных битов	—
Область таймера	—
Область счетчика	—
Область DM	—
Область EM, не содержащая банков	—
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—

Область	N
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–
Константы	От 0 до 1023
Регистры данных	–
Индексные регистры	–
Косвенная адресация с использованием индексных регистров	–

### Описание

Команда SBN(092) обозначает начало подпрограммы с указанным номером. Окончание подпрограммы обозначается командой RET(093).

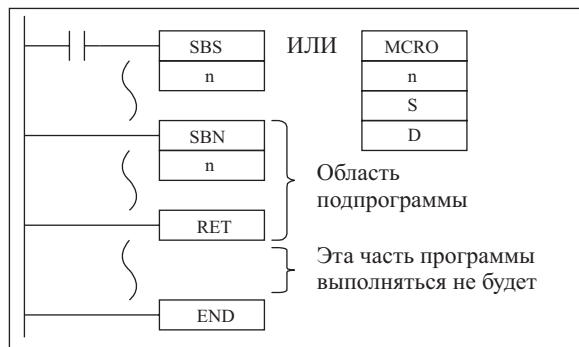
Область программы, начинающаяся первой командой SBN(092), является областью подпрограммы. Подпрограмма выполняется только тогда, когда она вызывается командой SBS(091) или MCRO(099).



### Меры предосторожности

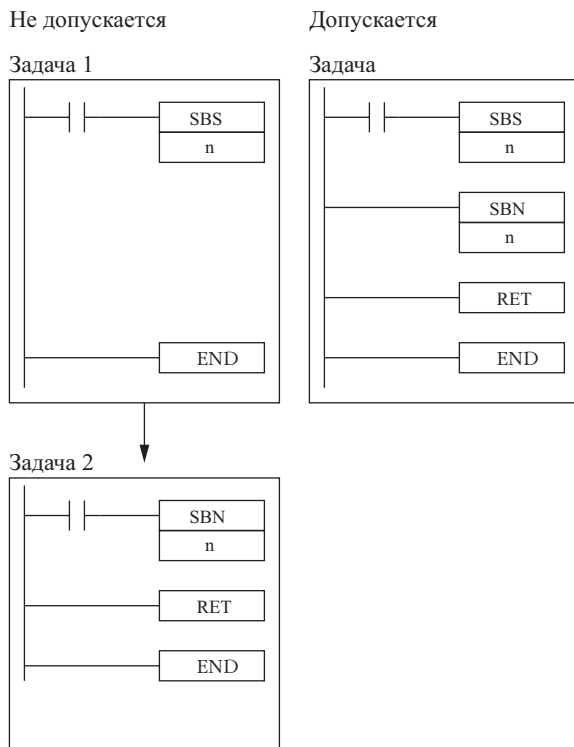
Когда подпрограмма не выполняется, ее команды используются в виде команд NPO(000).

Располагайте подпрограммы после основной программы, непосредственно после подпрограммы должна находиться команда END(001). Если часть основной программы располагается после области подпрограммы, эта часть программы будет игнорироваться.

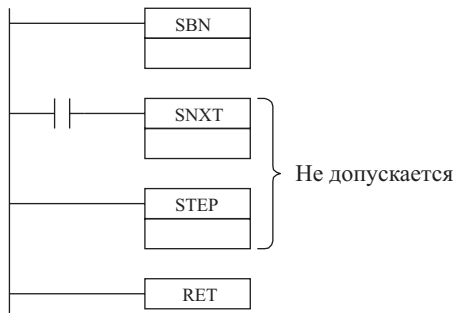


**Примечание:** Метод ввода номера подпрограммы *N* различен для *CX*-программатора и Пульта программирования. С помощью *CX*-программатора вводите номера от 0 до 1023 или от &0 до &127. С помощью Пульта программирования вводите номера от 0000 до 1023.

Убедитесь в том, что в каждой программе (задаче), где присутствует подпрограмма, введена соответствующая команда SBS(091) или MCRO(099). Подпрограмма, которая должна выполняться в одной задаче, не может вызываться из другой задачи. Существует возможность программирования подпрограммы в задаче прерывания.

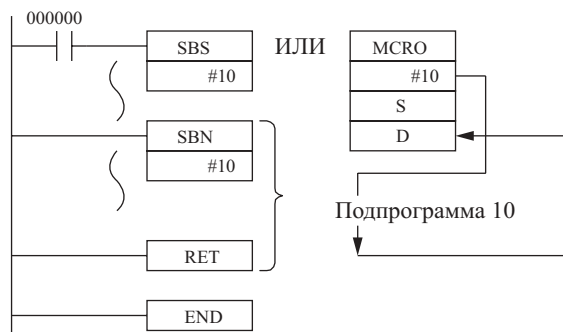


Шаговые команды, STEP(008) и SNXT(009) не могут выполняться в подпрограммах.



**Пример**

В следующем ниже примере, когда СІО 000000 находится в состоянии ON, осуществляется выполнение подпрограммы 10, после чего программа переходит к выполнению команды, следующей за командой SBS(091) или MCRO(099) (осуществляющей вызов подпрограммы 10).



**3-18-4 Команда окончания подпрограммы SUBROUTINE RETURN: RET(093)**

**Назначение**

Команда RET(093) обозначает окончание подпрограммы. Данная команда применяется совместно с командой SBN(092) для обозначения области подпрограммы.

**Символ релейно-контактной схемы**

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	RET(093)
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не используется	Не используется	См. примечание	Да

**Примечание:** Допускается использование только одной команды RET(093) в конце каждой из подпрограмм.

**Описание**

Команда SBN(092) обозначает начало подпрограммы, а команда RET(093) обозначает конец подпрограммы. Для более подробного ознакомления с работой подпрограмм обратитесь к разделу **3-18-3 «SUBROUTINE ENTRY: SBN(092)»**.

Когда по ходу выполнения программы, программа достигает команды RET(093), она автоматически переходит к выполнению команды, следующей за командой SBS(091) или MCRO(099), вызывающей подпрограмму. Перед завершением выполнения подпрограммы, вызванной командой MCRO(099), выходные данные из A604...A607 записываются в D...D+3.

**Меры предосторожности**

Когда подпрограмма не выполняется, ее команды выполняются в виде команды NOP(000).

**Пример**

Для ознакомления с примерами работы команды RET(093) обратитесь к разделу **3-18-3 «SUBROUTINE ENTRY: SBN(092)»**.

## 3-19 Команды управления прерываниями

## 3-19-1 Команда задания прерывания SET INTERRUPT MASK: MSKS(690)

**Назначение**

Команда MSKS(690) определяет процесс выполнения прерывания ввода/вывода или прерывания по расписанию.

Как задачи прерывания ввода/вывода, так и задачи прерывания по расписанию при первоначальном включении Программируемого контроллера маскированы (отключены). Команда MSKS(690) может применяться для маскирования (блокирования) и демаскирования (разблокирования) прерываний ввода/вывода, а также установления длительности интервалов для прерываний по расписанию.

**Символ релейно-контактной схемы**

MSKS(690)	
N	N: Идентификатор прерывания
S	S: Данные прерывания

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	MSKS(690)
	Выполняется один раз для дифференцирования вверх	@ MSKS(690)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****N: Идентификатор прерывания**

N идентифицирует прерывание. Номера от 0 до 3 обозначают номера Модулей ввода прерывания, а номера 4 и 5 обозначают прерывания по расписанию 2 и 3 соответственно.

**S: Данные прерывания**

Слово S содержит данные для маскирования прерываний ввода/вывода или данные об интервале для прерываний по расписанию.

N	Источник прерывания	Функция S
0	Модуль ввода прерывания 0	Биты от 0 до 7 слова S управляют задачами прерывания ввода/вывода от 100 до 107.
1	Модуль ввода прерывания 1	Биты от 0 до 7 слова S управляют задачами прерывания ввода/вывода от 108 до 115.
2	Модуль ввода прерывания 2	Биты от 0 до 7 слова S управляют задачами прерывания ввода/вывода от 116 до 123.
3	Модуль ввода прерывания 3	Биты от 0 до 7 слова S управляют задачами прерывания ввода/вывода от 124 до 131.
4	Прерывание 2 по расписанию	Слово S управляет задачами прерывания по расписанию 2 и 3. 0000: задача прерывания блокируется. 0001 270F: устанавливает величину интервала для выполнения прерывания (от 1 до 9999 T*). (T* устанавливается в начальных установках программируемого контроллера в единицах, равных 10 мсек. или 1 мсек.)
5	Прерывание 3 по расписанию	

**Спецификации операндов**

Область	N	S
Область ввода/вывода (область CIO)	—	CIO 0000...CIO 6143
Рабочая область	—	W000...W511
Область удержания бита (Holding bit area)	—	H000...H511
Область вспомогательных битов	—	A000...A447 A448...A959



Область	N	S
Область таймера	–	T0000...T4095
Область счетчика	–	C0000...C4095
Область DM	–	D00000...D32767
Область EM, не содержащая банков	–	E00000...E32767
Область EM, содержащая банки	–	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	–	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)
Константы	Только указанные значения	
Регистры данных	–	DR0...DR15
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров		,IR0...,IR15 -2048...+2047 ,IR0...-2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15(++)) ,(- -)IR0...,(- -)IR15

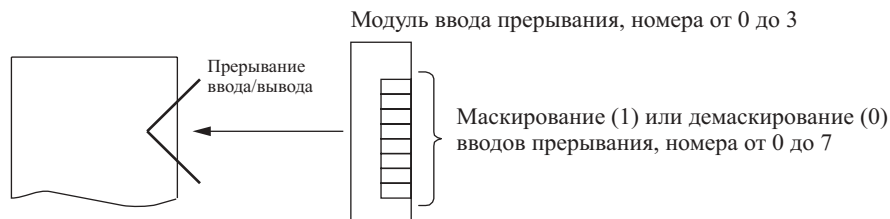
**Описание**

Команда MSKS(690) осуществляет управление прерываниями ввода/вывода и прерываниями по расписанию. Значение N идентифицирует прерывание.

**N= от 0 до 3**

Значения от 0 до 3 соответствуют номерам Модулей ввода прерывания от 0 до 3.

Биты от 0 до 7 слова S соответствуют номерам вводов прерывания от 0 до 7 в указанном Модуле. Команда MSKS(690) производит маскирование (блокирование) ввода прерывания, когда соответствующий бит находится в состоянии ON. Команда MSKS(690) демаскирует (разрешает) прерывание, когда соответствующий бит находится в состоянии OFF.



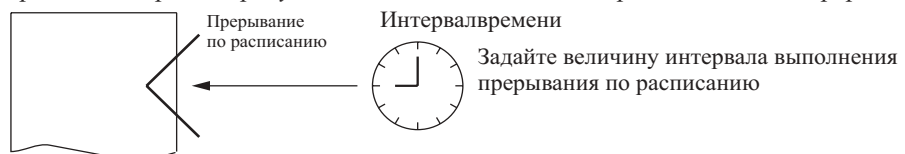
- Примечание:**
1. Команда MSKS(690) может использоваться для выполнения отдельной задачи прерывания ввода/вывода в отдельном цикле, обеспечивая блокирование задачи в других циклах.
  2. Номера распределяются Модулям ввода прерывания по порядку их установки, слева направо.

**N= 4 или 5**

Значения 4 и 5 соответствуют прерываниям по расписанию 2 и 3.

Когда N равно 4 или 5, содержание слова S либо блокирует задачу прерывания (S=0000), либо задает выполнение задачи прерывания в указанном интервале времени (от S=0001 до S=270F). Единицы времени для задания интервала устанавливаются в начальных установках Программируемого контроллера (00: 10 мсек., 01: 1.0 мсек.), поэтому диапазонами значений для интервала выполнения прерывания являются от 10 мсек. до 99.99 сек и от 1 мсек. до 9.999 сек.

- Примечание:** Не устанавливайте величину интервала для выполнения прерывания по расписанию меньше чем 10 мсек. Кроме того, убедитесь в том, что длительность интервала превышает время, требуемое для выполнения запланированной задачи прерывания.



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение N выходит за пределы от 0 до 5. Переводится в состояние ON, когда S выходит за пределы допустимых значений от 0000 до 00FF, а значение N - от 0 до 3. Переводится в состояние ON, когда S выходит за пределы допустимых значений от 0000 до 270F, а значение N равно 4 или 5. OFF в других случаях.

В следующей ниже таблице приводятся флаги вспомогательной области, относящиеся к данной команде.

Наименование	Адрес	Функционирование
Флаг ошибки задачи прерывания	A40213	Переводится в состояние ON в следующих случаях: Задача прерывания длительностью более 10 мсек. выполнена в процессе выполнения регенерации ввода/вывода из Специального модуля C200H или Slave-панели удаленного ввода/вывода. В задаче прерывания выполнена команда IOFR(097) без блокирования периодической регенерации специального ввода/вывода.
Флаг причины ошибки задачи прерывания	A42615	Обозначает номер ошибки (1 или 2) задачи прерывания.
Номер задачи прерывания, где произошла ошибка	A42600...A42611	Для ошибки 1: Указывает номер ошибочной задачи прерывания. Для ошибки 2: Указывает номер Специального модуля, где произведена множественная регенерация ввода/вывода.

**Меры предосторожности**

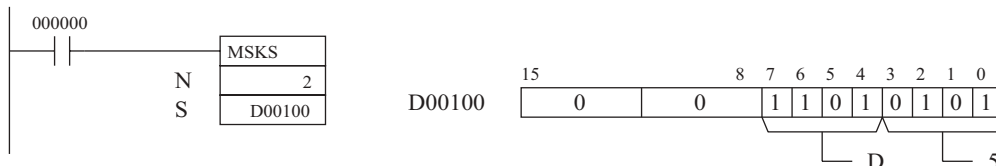
Поддерживаются вводы прерывания от обычных Модулей ввода прерывания CS1 и Модулей ввода прерывания C200H. Вводы прерываний от Встроенных плат и Специальных модулей Не поддерживается. Прерывания обладают различными уровнями приоритета. Прерываниям при отключении питания присваивается наиболее высокий приоритет, затем следуют прерывания по расписанию, и, наконец, прерывания ввода/вывода. Прерывания ввода/вывода, имеющие меньшие номера обладают приоритетом над прерываниями, имеющими более высокие номера.

В случае, когда в системе работают Специальные модули C200H или Slave-панели удаленного ввода/вывода SYSMAC BUS, убедитесь в том, что выполнение задачи прерывания не требует более чем 10 мсек. Если задача прерывания, занимающая более чем 10 мсек., выполняется в процессе регенерации ввода/вывода из Специального модуля или Slave-панели, определяется допустимая ошибка и флаг ошибки задачи прерывания (A40213) переводится в состояние ON.

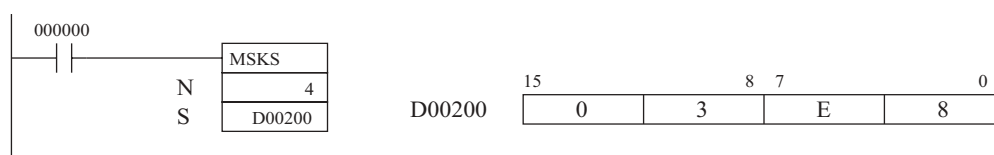
Когда для регенерации ввода/вывода Специального модуля команда IOFR(097) выполняется в задаче прерывания, периодическая регенерация ввода/вывода должна отключаться в начальных установках программируемого контроллера. Если в такой ситуации периодическая регенерация остается в действии, команда IOFR(097) может быть выполненной в процессе выполнения периодической регенерации, приведя к определению ошибки вследствие двойной регенерации и переводу флага ошибки задачи прерывания (A40213) в состояние ON.

**Примеры**

Когда в следующем ниже примере CIO 000000 переводится в состояние ON, команда MSKS(690) производит демаскирование вводов прерывания 1, 3 и 5 Модуля ввода прерывания №2.



Когда в следующем ниже примере CIO 000001 переводится в состояние ON, команда MSKS(690) задает интервал, равный 10 сек., для выполнения прерывания 2 по расписанию. (В данном случае в начальных установках программируемого контроллера установлены единицы времени, равные 10 мсек.)

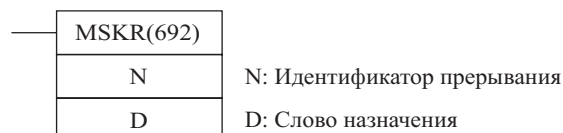


### 3-19-2 Команда чтения данных управления прерыванием READ INTERRUPT MASK: MSKR(692)

#### Назначение

По команде MSKR(692) производится чтение установок, определяющих процесс управления прерыванием, т.е. установок, заданных командой MSKS(690).

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	MSKR(692)
	Выполняется один раз для дифференцирования вверх	@ MSKR(692)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

##### N: Идентификатор прерывания

N идентифицирует прерывание. Номера от 0 до 3 обозначают номера Модулей ввода прерывания, а номера 4 и 5 обозначают прерывания по расписанию 2 и 3 соответственно.

##### D: Слово назначения

Данные маскирования прерываний ввода/вывода или данные об интервале для выполнения прерываний по расписанию записываются в слово D.

N	Источник прерывания	Содержание D
0	Модуль ввода прерывания 0	Биты от 0 до 7 показывают состояние прерываний ввода/вывода от 100 до 107.
1	Модуль ввода прерывания 1	Биты от 0 до 7 показывают состояние прерываний ввода/вывода от 108 до 115.
2	Модуль ввода прерывания 2	Биты от 0 до 7 показывают состояние прерываний ввода/вывода от 116 до 123.
3	Модуль ввода прерывания 3	Биты от 0 до 7 показывают состояние прерываний ввода/вывода от 124 до 131.
4	Прерывание 2 по расписанию	Слово D указывает состояние прерываний по расписанию 2 и 3. 0000: задача прерывания блокируется. 0001...270F: Длительность интервала для выполнения прерывания (от 1 до 9999 T*). (T* устанавливается в начальных установках программируемого контроллера в единицах, равных 10 мсек. или 1 мсек.)
5	Прерывание 3 по расписанию	

#### Спецификации операндов

Область	N	S
Область ввода/вывода (область CIO)	—	CIO 0000...CIO 6143
Рабочая область	—	W000...W511

Область	N	S
Область удержания бита (Holding bit area)	–	H000...H511
Область вспомогательных битов	–	A000...A447 A448...A959
Область таймера	–	T0000...T4095
Область счетчика	–	C0000...C4095
Область DM	–	D00000...D32767
Область EM, не содержащая банков	–	E00000...E32767
Область EM, содержащая банки	–	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	–	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)
Константы	Только указанные значения	
Регистры данных	–	DR0...DR15
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров		,IR0...,IR15 -2048...+2047,IR0...-2048...+2047,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15

### Описание

По команде MSKR(692) осуществляется чтение установок, заданных командой MSKS(690). Если указан Модуль ввода прерывания, по команде MSKR(692) производится чтение текущего состояния маскирования прерывания для данного Модуля.

**N= от 0 до 3**

Значения от 0 до 3 соответствуют номерам Модулей ввода прерывания от 0 до 3.

Биты от 0 до 7 слова D соответствуют номерам вводов прерывания от 0 до 7 в указанном Модуле. Если бит находится в состоянии ON, соответствующий ввод прерывания заблокирован. Если бит находится в состоянии OFF, ввод соответствующего прерывания разрешен.

**N= 4 или 5**

Значения 4 и 5 соответствуют прерываниям 2 и 3 по расписанию.

Когда N равно 4 или 5, содержание D показывает величину интервала, установленного для выполнения прерывания. Значение, равное 0000, обозначает блокирование прерывания. Единицы времени для задания интервала устанавливаются в начальных установках Программируемого контроллера (00: 10 мсек., 01: 1.0 мсек.), поэтому диапазонами значений интервала для выполнения прерывания являются от 10 мсек. до 99.99 сек и от 1 мсек. до 9.999 сек.

### Флаги

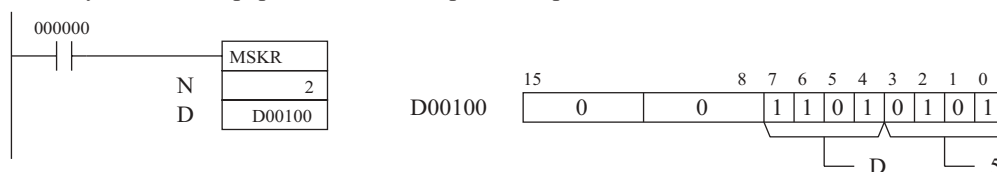
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение N выходит за пределы от 0 до 5. OFF в других случаях.

### Меры предосторожности

Команда MSKR(692) может выполняться в основной задаче или в задачах прерывания.

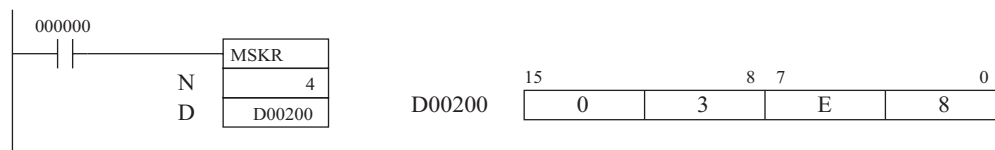
### Примеры

Когда в следующем ниже примере CIO 000000 переводится в состояние ON, по команде MSKR(692) производится чтение текущего состояния управления прерыванием для Модуля ввода прерывания №2. В данном случае вводы прерывания 1, 3 и 5 разблокированы.



Когда в следующем ниже примере CIO 000100 переводится в состояние ON, по команде MSKR(692) производится чтение установки для прерывания 2 по расписанию. В данном случае заданная величина прерывания

вания равна 1000 (3E8 шестн.), что эквивалентно 10 сек, если в начальных установках Программируемого контроллера единицы времени установлены в значение, равное 10 мсек.



### 3-19-3 Команда сброса прерывания CLEARINTERRUPT: CLI(691)

#### Назначение

Команда CLI(691) осуществляет сброс или сохранение записанных вводов прерывания (ввода/вывода), или установку времени для первого прерывания по расписанию.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	CLI(691)
	Выполняется один раз для дифференцирования вверх	@ CLI(691)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

##### N: Идентификатор прерывания

N идентифицирует прерывание. Номера от 0 до 3 обозначают номера Модулей ввода прерывания, а номера 4 и 5 обозначают прерывания по расписанию 2 и 3 соответственно.

##### S: Данные прерывания

Слово S содержит данные о сбросе/сохранении данных прерываний ввода/вывода, или данные об интервале для прерываний по расписанию.

N	Источник прерывания	Функция S
0	Модуль ввода прерывания 0	Биты от 0 до 7 слова S управляют задачами прерывания от 100 до 107.
1	Модуль ввода прерывания 1	Биты от 0 до 7 слова S управляют задачами прерывания от 108 до 115.
2	Модуль ввода прерывания 2	Биты от 0 до 7 слова S управляют задачами прерывания от 116 до 123.
3	Модуль ввода прерывания 3	Биты от 0 до 7 слова S управляют задачами прерывания от 124 до 131.
4	Прерывание 2 по расписанию	Слово S устанавливает время выполнения первой задачи прерывания по расписанию. 0000: задача прерывания блокируется. 0001...270F: устанавливает величину интервала для выполнения прерывания (от 1 до 9999 T*). (T* устанавливается в начальных установках программируемого контроллера в единицах, равных 10 мсек. или 1 мсек.)
5	Прерывание 3 по расписанию	

### Спецификации операндов

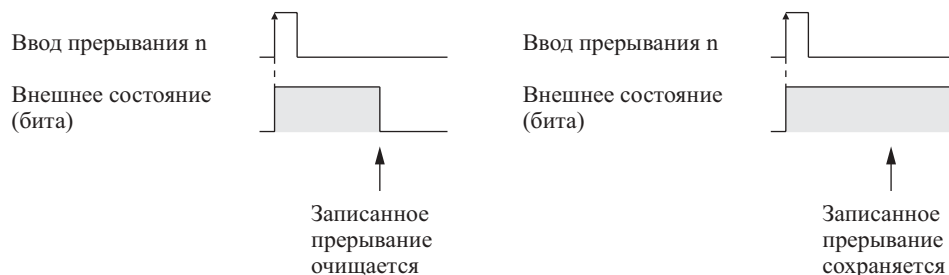
Область	N	S
Область ввода/вывода (область CIO)	–	CIO 0000...CIO 6143
Рабочая область	–	W000...W511
Область удержания бита (Holding bit area)	–	H000...H511
Область вспомогательных битов	–	A000...A447 A448...A959
Область таймера	–	T0000...T4095
Область счетчика	–	C0000...C4095
Область DM	–	D00000...D32767
Область EM, не содержащая банков	–	E00000...E32767
Область EM, содержащая банки	–	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	–	@D00000...@D32767 @E00000...@E32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)
Константы	–	DR0...DR15
Регистры данных	Только указанные значения	
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров	–	,IR0...IR15 -2048...+2047 ,IR0...-2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-(-)IR15

### Описание

В зависимости от значения N, по команде CLI(691) осуществляется либо очистка указанных записанных прерываний ввода/вывода, либо установка времени до выполнения первого прерывания по расписанию. N= от 0 до 3

Значения от 0 до 3 соответствуют номерам Модулей ввода прерывания от 0 до 3.

Биты от 0 до 7 слова S соответствуют номерам вводов прерывания от 0 до 7 в указанном Модуле. Команда CLI(691) производит сброс записанного ввода прерывания, когда соответствующий бит слова S находится в состоянии ON. Записанный ввод прерывания сохраняется, если соответствующий бит слова S находится в состоянии OFF.



Если задача прерывания ввода/вывода выполняется, и принимается прерывание с другим номером, этот номер прерывания записывается внутри Модуля. Записанные прерывания ввода/вывода выполняются позднее в порядке присвоенного им приоритета (начиная с меньшего номера к большему номеру.) Команда CLI(691) может использоваться для сброса записанных прерываний перед их выполнением.

- Примечание:**
1. Команда MSKS(690) может использоваться для выполнения отдельной задачи прерывания ввода/вывода в отдельном цикле, обеспечивая блокирование этой задачи в других циклах.
  2. Номера распределяются Модулям ввода прерывания по порядку их установки, слева направо.

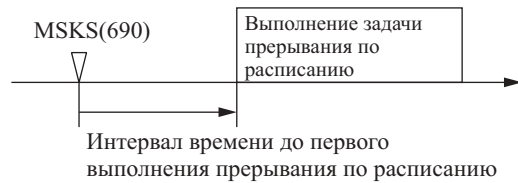
N= 4 или 5

Значения 4 и 5 соответствуют прерываниям по расписанию 2 и 3.

Когда N равно 4 или 5, содержание S устанавливает время выполнения первой задачи прерывания по расписанию после выполнения команды MSKS(690).

Диапазон установки интервала – от 0000 до 270F (от 0 до 9999). Единицы времени для задания интервала устанавливаются в начальных установках Программируемого контроллера (00: 10 мсек., 01: 1.0 мсек.), поэтому в действительности диапазонами значений интервала для выполнения прерывания являются от 10 мсек. до 99.99 сек и от 1 мсек. до 9.999 сек.

**Примечание:** Для выполнения первого прерывания по расписанию задавайте интервал, равный 10 мсек. и более.



### Флаги

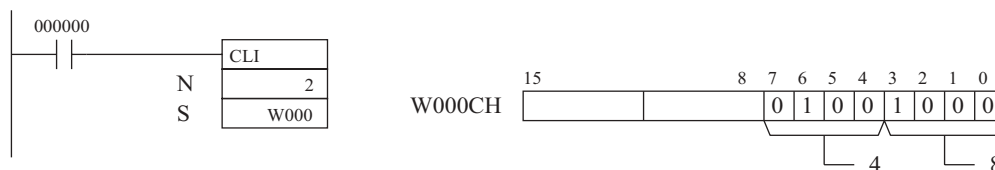
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение N выходит за пределы от 0 до 5. Переводится в состояние ON, когда S выходит за пределы допустимых значений от 0000 до 00FF, а значение N - от 0 до 3. Переводится в состояние ON, когда S выходит за пределы допустимых значений от 0000 до 270F, а значение N равно 4 или 5. OFF в других случаях.

### Меры предосторожности

Количество вводов прерываний, которое может быть записано, не ограничено. Возможна запись всех вводов прерывания ввода/вывода, однако новое прерывание будет игнорировано, если его запись уже произведена. Более того, записанное прерывание не может сбрасываться до завершения выполнения соответствующей ему задачи прерывания. Поэтому если новый ввод прерывания принимается в процессе выполнения соответствующей ему задачи прерывания, это прерывание будет игнорировано. Поддерживаются вводы прерываний от обычных Модулей ввода прерывания CS1 и Модулей ввода прерывания S200H. Вводы прерываний от Встроенных плат и Специальных модулей не поддерживаются. Номера прерываний распределяются по порядку во всех Модулях ввода прерывания. Прерывания обладают различными уровнями приоритета. Прерываниям при отключении питания присваивается наиболее высокий приоритет, затем следуют прерывания по расписанию, и, наконец, прерывания ввода/вывода. Прерывания ввода/вывода, имеющие меньшие номера, обладают преимуществом над прерываниями, имеющими более высокие номера.

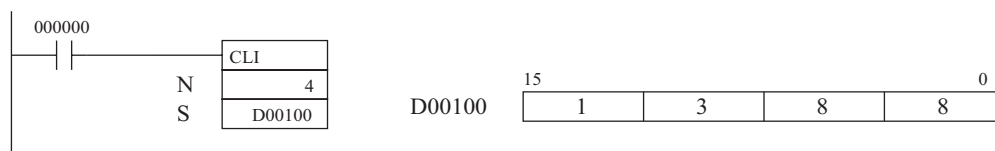
### Примеры

Когда в следующем ниже примере CIO 000000 переводится в состояние ON, команда CLI(692) производит очистку записанных прерываний для вводов прерываний 3 и 6 Модуля ввода прерывания №2.



Когда в следующем ниже примере CIO 000001 переводится в состояние ON, команда CLI(692) задает интервал от 2 до 50 мсек. до первого выполнения прерывания по расписанию.

В данном случае единицы для задания интервала установлены в начальных установках Программируемого контроллера в значение, равное 10 мсек.





## 3-19-4 Команда блокирования прерываний DISABLE INTERRUPTS: DI(693)

**Назначение**

Команда DI(693) отменяет выполнение всех задач прерывания, за исключением задачи прерывания при отключении питания.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	DI(693)
	Выполняется один раз для дифференцирования вверх	@ DI(693)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Не используется

**Описание**

Команда DI(693) выполняется из основной программы для временной отмены выполнения всех задач прерывания, за исключением задачи прерывания при отключении питания (т.е. отмены прерываний ввода/вывода, прерываний по расписанию, и внешних прерываний).

По команде DI(693) выполнение всех задач прерывания отменяется до выполнения команды EI(694), после которой выполнение всех задач прерывания разрешается.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда команда DI(693) выполняется из задачи прерывания. OFF в других случаях.

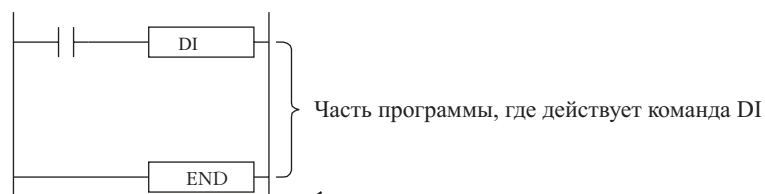
**Меры предосторожности**

Выполнение всех задач прерывания отменяется до выполнения команды EI(694).

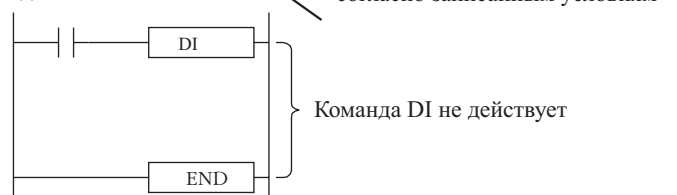
Команда DI(693) не может выполняться из задачи прерывания.

Команда DI(693) не может выполняться более чем для одной циклической задачи. Для отмены выполнения прерываний более чем в одном цикле, вводите данную команду в каждую из циклических задач. Любые прерывания, возникающие при выполнении задачи длительностью в один цикл, будут выполняться после завершения текущей задачи, если их выполнение не отменено командой CLI(691), как показано на следующем ниже рисунке.

Задача №0



Задача №1

**Примеры**

Когда в следующем ниже примере СЮ 000000 переводится в состояние ON, команда DI(693) производит отмену выполнения всех задач прерывания, за исключением задачи прерывания при отключении питания.



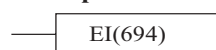


### 3-19-5 Команда разрешения выполнения прерываний ENABLE INTERRUPTS: EI(694)

#### Назначение

Команда EI(694) разрешает выполнение всех задач прерывания, выполнение которых отменялось командой DI(693).

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	EI(694)
	Выполняется один раз для дифференцирования вверх	@ EI(694)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Не используется

#### Описание

Команда EI(694) выполняется из основной программы для временного разрешения выполнения всех задач прерывания, выполнение которых отменялось командой DI(693). Команда DI(693) отменяет выполнение всех задач прерывания, за исключением задачи прерывания при отключении питания (т.е. отменяет выполнение прерываний ввода/вывода, прерываний по расписанию, и внешних прерываний).

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда команда EI(694) выполняется из задачи прерывания. OFF в других случаях.

#### Меры предосторожности

Команда EI(694) не требует условия выполнения. При выполнении данной команды условие выполнения всегда находится в состоянии ON. Команда EI(694) разрешает выполнение всех задач прерывания, которые отменялись командой DI(693).

Данная команда не может разрешить выполнение прерываний ввода/вывода, выполнение которых не было разрешено командой MSKS(690), или задавать прерывания по расписанию, которые не были заданы командой MSKS(690).

Команда EI(694) не может выполняться из задачи прерывания.

#### Примеры

В следующем ниже примере команда EI(694) разрешает выполнение всех задач прерывания, выполнение которых было отменено командой DI(693).



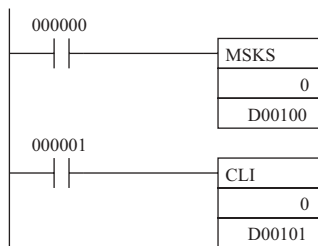
### 3-19-6 Резюме по управлению прерываниями

Команды управления прерываниями осуществляют управление прерываниями, а также чтение установок для прерываний ввода/вывода и прерываний по расписанию. (Команды DI(603) и EI(694) осуществляют управление не только прерываниями ввода/вывода и прерываниями по расписанию, но и управление внешними прерываниями.)

Команды, осуществляющие управление отдельными прерываниями, содержат операнд N, идентифицирующий источник прерывания. Номера от 0 до 3 обозначают Модули ввода прерывания, а номера 4 и 5 указывают прерывания 2 и 3 по расписанию.

#### Выполнение прерываний ввода/вывода (N=0...3)

Прерывание ввода/вывода вызывается входным сигналом, приходящим из Модуля ввода прерывания. К программируемому контроллеру может подключаться до 4-х Модулей ввода прерывания. Номера от 0 до 3 распределяются Модулям согласно их расположению в Программируемом контроллере, слева направо. Следующий ниже пример программы демонстрирует работу команд MSKS(690) и CLI(691), когда эти команды используются для управления прерываниями ввода/вывода.



D00100	0	0	F	0
D00101	0	0	F	2

#### Выполнение команды MSKS(690)

При первоначальном включении Программируемого контроллера задачи прерывания ввода/вывода и задачи прерывания по расписанию маскированы (блокированы). Команда MSKS(690) может использоваться для демаскирования и маскирования прерываний ввода/вывода или для установки интервалов времени выполнения прерываний по расписанию.

В данном примере, команда MSKS(690) использует содержание слова D00100 для разблокирования вводов 0...3 и для блокирования вводов 4...7 из Модуля ввода прерывания 0.

	F				0			
Ввод прерываний из Модуля 0	7	6	5	4	3	2	1	0
Установки для блокирования прерываний	1	1	1	1	0	0	0	0

1= Маскирование (блокирование). 0= Демаскирование (разблокирование).

Когда ввод прерывания 3 переходит из состояния OFF в состояние ON, выполнение основной программы прерывается, после чего начинается выполнение задачи прерывания 3 ввода/вывода (задача прерывания 103). Выполнение основной программы возобновляется в точке прерывания после завершения задачи прерывания 3 ввода/вывода.

#### Приоритет задачи прерывания ввода/вывода

В случае, когда два и более прерываний принимается одновременно, выполнение прерываний производится согласно номерам прерываний, начиная с меньшего номера (100...131).

Модуль	Задачи прерывания
Модуль ввода прерывания 0	Вводы от 0 до 7 соответствуют задачам прерывания от 100 до 107.
Модуль ввода прерывания 1	Вводы от 0 до 7 соответствуют задачам прерывания от 108 до 115.
Модуль ввода прерывания 2	Вводы от 0 до 7 соответствуют задачам прерывания от 116 до 123.
Модуль ввода прерывания 3	Вводы от 0 до 7 соответствуют задачам прерывания от 124 до 131.

Когда в процессе выполнения задачи прерывания принимается дополнительное количество вводов прерывания, записанные прерывания выполняются в порядке их приоритета после завершения выполнения текущей задачи прерывания.

Задача прерывания по расписанию при ее возникновении имеет приоритет над задачами прерывания ввода/вывода.

#### Выполнение команды CLI(691)

В случае приема ввода прерывания в процессе выполнения другой задачи прерывания ввода/вывода, номер принятого ввода прерывания запоминается, и прерывание выполняется после завершения выполнения текущей задачи и задач, имеющих более высокий приоритет. Команда CLI(691) может использоваться для очистки записанных прерываний до их выполнения, однако не может использоваться для сброса выполняемых задач прерывания.

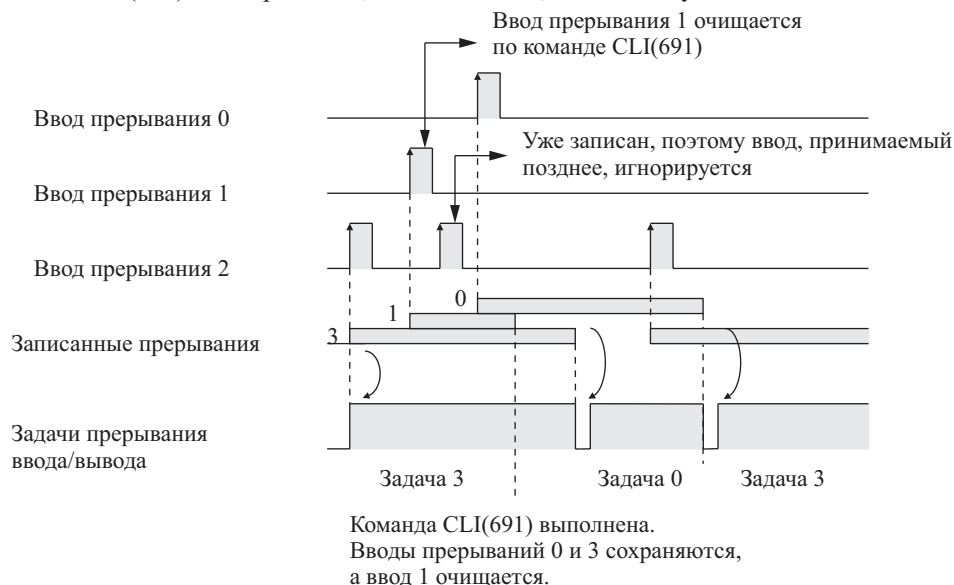
В данном примере команда CLI(691) использует содержание D00101 для очистки всех записанных прерываний из Модуля ввода прерываний 0, за исключением вводов 0, 2 и 3.

Рисунок.

	F				2			
Ввод прерываний из Модуля 0	7	6	5	4	3	2	1	0
Установки для блокирования прерываний	1	1	1	1	0	0	1	0

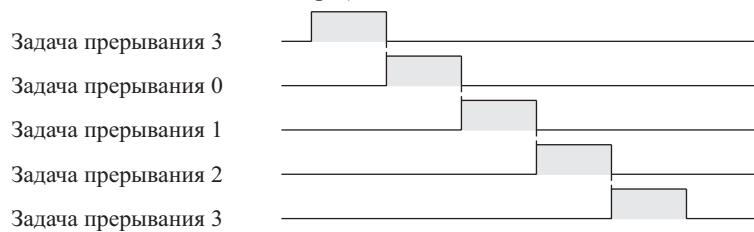
1= Очистка записанного ввода. 0 = Сохранение записанного ввода.

После завершения выполнения задачи прерывания 3, записанные прерывания выполняются согласно присвоенному приоритету. Вследствие того, что ввод прерывания 0 был записан, задача прерывания 0 (задача прерывания 100) будет выполняться после завершения выполнения задачи 3. Ввод прерывания 1 по команде CLI(691) не сохраняется, следовательно, этот ввод будет очищен.



Когда вводы прерываний 0...3 переводятся в состояние ON, а команда CLI(691) не выполняется, все вводы будут записаны и после завершения выполнения задачи 3 все задачи прерывания выполнены согласно

их приоритету. (Задачи прерывания выполняются по порядку, согласно присвоенному им приоритету, начиная с наименьшего номера.)



**Примечание:**

1. Применение команды `CLI(691)` не во всех случаях обязательно.
2. Когда команда `CLI(691)` не выполняется, все вводы прерываний, получаемые в процессе выполнения задачи прерывания, будут записаны. Если записанное прерывание принимается вторично, полученное позднее прерывание игнорируется.
3. Когда записывается два и более прерываний ввода/вывода, они выполняются в порядке их приоритета. При этом порядок, в котором они были записаны, значения не имеет.

### Выполнение прерывания по расписанию (N = 4 или 5)

Прерывания по расписанию повторяются через заданные промежутки времени, определяемые командой `MSKS(690)`, и не зависят от длительности цикла Программируемого контроллера. Номера N 4 и 5 соответствуют номерам 2 и 3 прерываний по расписанию.

#### Выполнение прерывания по расписанию

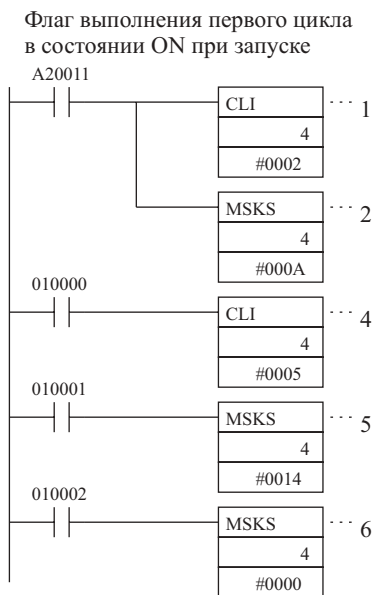
Основные особенности выполнения прерывания по расписанию перечислены ниже.

**1,2,3...**

1. При первоначальном включении Программируемого контроллера прерывания по расписанию маскированы (блокированы).
2. С помощью команды `CLI(691)` задайте время выполнения первого прерывания по расписанию (после выполнения команды `MSKS(690)`). В случае, когда время выполнения первого прерывания по расписанию не устанавливается, это прерывание может произойти в непредсказуемое время.
3. Установка интервала для прерывания по расписанию и выполнение прерывания.
  - С помощью команды `MSKS(690)` задайте интервал для выполнения прерывания.
  - После выполнения команды `MSKS(690)` и истечения времени выполнения первого прерывания по расписанию (заданного командой `CLI(691)`), выполнение текущей задачи прерывается и начинается выполнение задачи прерывания по расписанию.
  - По мере выполнения задачи прерывания и достижения команды `END(001)`, выполнение программы возобновляется, начиная с точки, в которой она была остановлена для выполнения прерывания по расписанию.
  - По истечению заданного интервала времени выполнение программы будет прервано вторично для следующего выполнения задачи прерывания по расписанию. Таким образом, задача прерывания по расписанию будет выполняться периодически, до тех пор, пока не будет заблокирована.
4. Блокирование задачи прерывания по расписанию.
  - Задача прерывания по расписанию может блокироваться (отключаться) посредством задания командой `MSKS(690)` интервала для выполнения прерывания, равного 0000.
  - При разрешении выполнения задачи прерывания по расписанию, перед выполнением команды `MSKS(690)` не забудьте вновь задать интервал для выполнения задачи при помощи команды `CLI(691)`.

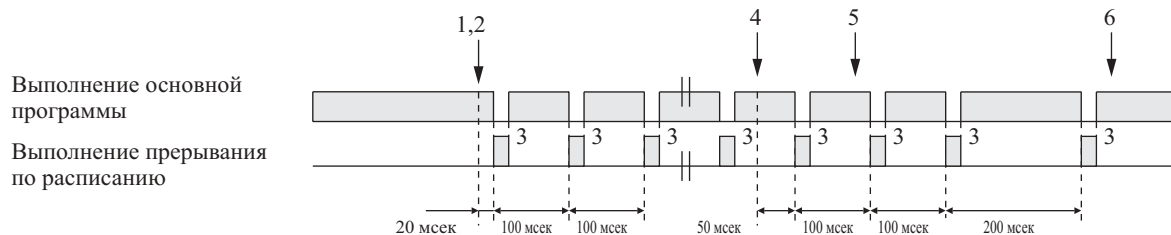
#### Выполнение прерывания по расписанию

В следующем ниже примере, единицы для задания интервала времени установлены в начальных установках Программируемого контроллера в значение, равное 10 мсек.



- 1,2,3...**
1. С помощью команды CLI(691) время выполнения первого прерывания по расписанию установлено в значение, равное 20 мсек.
  2. Интервал для выполнения прерывания по расписанию установлен в значение, равное 100 мсек. и выполнение прерывания 2 разрешено командой MSKS(690).
  3. Спустя 20 мсек. после выполнения команды MSKS(690) выполняется первое прерывание по расписанию, затем оно выполняется через каждые 100 мсек.
  4. После начала выполнения прерывания по расписанию, время выполнения следующего прерывания может изменяться при помощи команды CLI(691), однако это изменение будет действительно только один раз.
  5. После начала выполнения прерывания по расписанию, интервал выполнения прерываний может изменяться при помощи команды MSKS(690). В данном случае длительность интервала для выполнения прерывания изменяется от значения, равного 100 мсек., до значения, равного 200 мсек.
  6. Выполнение прерываний по расписанию отменяется путем изменения длительности интервала на значение, равное 0000, при помощи команды MSKS(690).

Ниже приводится диаграмма, поясняющая работу предложенной выше программы.

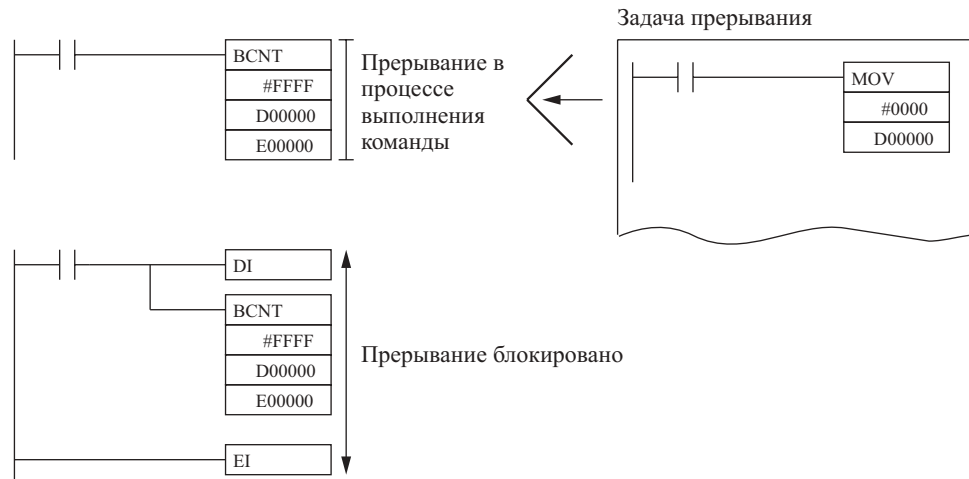


### Меры предосторожности

Убедитесь в том, что длительность интервала между выполнением прерываний по расписанию превышает время, требуемое для выполнения задачи прерывания по расписанию. Если интервал выполнения прерывания мал, задача прерывания будет выполняться длительное время, что приведет к определению ошибки вследствие превышения длительности цикла. (Длительная задача прерывания по расписанию может значительно увеличить общее время выполнения программы.)

Прерывание по расписанию выполняется по истечению заданного интервала времени, к этому времени необходимо добавить время выполнения одной команды. Обычно, время, требуемое для выполнения одной команды, пренебрежимо мало, однако, при выполнении команды, имеющей длительное время выполнения, это может стать причиной определения ошибки. Кроме того, выполнение длительной команды может стать причиной ошибки в работе таймеров (TIM и TIMH), а также при выполнении процедуры отслеживания данных. Особенно будьте внимательны, если в начальных установках Программируемого контроллера единицы времени установлены в значение, равное 0.5 или 1 мсек.

Прерывания принимаются даже в том случае, когда выполняется только одна команда. Следовательно, если прерывание принимается в процессе выполнения команды, требующей длительного времени выполнения, существует возможность получения ошибочных данных, так как и задача прерывания и команда могут обращаться к одним и тем же данным. Поэтому в таком случае используйте команды DI(690) и EI(691) для блокирования и разрешения выполнения прерывания.





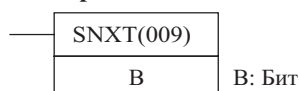
## 3-20-1 Команды шаговых программ STEP DEFINE and STEP START: STEP(008)/ SNXT(009)

**Назначение**

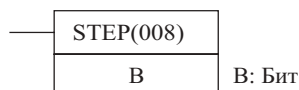
Команда SNXT(009) располагается непосредственно перед командой STEP(008) и управляет выполнением шага посредством перевода указанного бита управления в состояние ON. Если непосредственно перед командой SNXT(009) располагается другой шаг, эта команда осуществляет также перевод бита управления предшествующего шага в состояние OFF.

Команда STEP(008) располагается непосредственно после команды SNXT(009) и перед выполнением каждого процесса. Команда обозначает запуск каждого из процессов и указывает бит управления процессом.

Кроме того, эта команда располагается в конце области программирования шагов после последней команды SNXT(009) для обозначения конца области программирования шагов. Когда такая команда появляется в конце области программирования шагов, она не использует бит управления.

**Символ релейно-контактной схемы**

Когда определяется начало шага, то бит управления специфицируется следующим образом:



Когда определяется конец шага, то бит управления не специфицируется:

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	SNXT(009) STEP(008)
	Выполняется один раз для дифференцирования вверх	Не поддерживается
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков программы	Области шагов программы	Подпрограммы	Задачи прерывания
Не используется	Да	Не используется	Не используется

**Спецификации операндов**

Область	N	S
Область ввода/вывода (область CIO)	—	
Рабочая область	W00000...W51115	
Область удержания бита (Holding bit area)	—	
Область вспомогательных битов	—	
Область таймера	—	
Область счетчика	—	
Область DM	—	
Область EM, не содержащая банков	—	
Область EM, содержащая банки	—	
Адреса DM/EM памяти при косвенном указании в двоичном коде	—	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—	
Константы	—	
Регистры данных	—	
Индексные регистры	—	



Область	N	S
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(++ ,(- -)IR0...,-(- -)IR15	

**Описание****SNXT(009)**

Существует три пути использования команды SNXT(009):

- 1,2,3...**
1. Для запуска шаговой программы.
  2. Для перехода к следующему биту управления.
  3. Для завершения выполнения шаговой программы.

Область программирования шагов начинается с первой команды STEP(008) (которая всегда использует бит управления) до последней команды STEP(009) (которая никогда не использует бит управления.)

- Запуск выполнения шаговой программы

Команда SNXT(009) всегда помещается в начале области программирования шагов для запуска шаговой программы. Команда переводит бит управления В, указанный для следующей команды STEP(008), в состояние ON, и переходит к выполнению шага В (т.е. к выполнению всех команд после команды STEP(008)В). Для команды STEP(008), начинающей выполнение шаговой программы, должно применяться дифференцированное условие выполнения, или программа будет выполняться только в одном цикле.

- Переход к выполнению следующего шага

Когда команда SNXT(009) встречается в середине области программирования шага (в середине шаговой программы), она используется для перехода к выполнению следующего шага. Эта команда переводит бит управления предшествующего шага в состояние OFF, и переводит бит управления следующего шага (В) в состояние ON, запуская выполнение следующего шага (т.е. запуская выполнение всех команд после команды STEP(008) В).

- Окончание области программирования шагов

Когда команда SNXT(009) располагается в самом конце области программирования шагов, она завершает выполнение шаговой программы и переводит в состояние OFF предшествующий бит управления. Указываемый бит управления (В) является «холостым битом». Тем не менее, этот бит должен быть переведен в состояние ON, поэтому указывайте бит, который при его использовании не вызовет каких либо проблем.

**STEP(008)**

В зависимости от расположения, и от того, указан или не указан бит управления, команда STEP(008) функционирует двумя способами.

- 1,2,3...**
1. Запускает выполнение указанного шага.
  2. Завершает область программирования шагов.

- Запуск выполнения шага

Команда STEP(008) помещается в начале каждого из шагов и содержит операнд (В), который используется в качестве бита управления.

Бит управления В переводится в состояние ON командой SNXT(009), после чего начинается выполнение команды, находящейся непосредственно после команды STEP(008). Одновременно с началом выполнения шага флаг шага (A20012) переводится в состояние ON.

После завершения выполнения первого шага, его выполнение возобновляется и повторяется до наступления условий перехода к следующему шагу, т.е. до момента, когда по команде SNXT(009) бит управления следующего шага (в следующей команде STEP(009)) переводится в состояние ON. Когда команда SNXT(009) переводит бит управления следующего шага в состояние ON, бит управления текущей команды (В) переустанавливается (OFF) и шаг, управляемый этим битом, блокируется.

Управление выводами и командами в шаге изменяется в зависимости от состояния бита управления В. (Состояние бита управления контролируется командой SNXT(009)). Когда бит управления переводится в состояние OFF, команды, содержащиеся в шаге, обнуляются и шаг блокируется. Обратимся к следующей таблице.

Состояние бита управления	Работа
ON	Команды, содержащиеся в шаге, выполняются обычным путем.
ON→OFF	Биты и команды в шаге заблокированы, как показано в следующей таблице.
OFF	Все команды в шаге выполняются как команды NOP.

**Состояние блокирования (IL)**

Выход команды	Состояние
Биты, указанные для OUT, OUT NOT	Все в состоянии OFF
Команды управления таймерами TIM, TIMH, TMHN и TIML	Текущее значение (PV)
	Флаг завершения
Биты или слова, указанные для других команд (см. примечание.)	Удерживает предшествующее состояние (тем не менее, команды не выполняются).

**Примечание:** Обозначает прочие команды, такие как функциональные коды TTIM(087), MTIM(543), SET, RESET, CNT, CNTR(012), SFT(010) и KEEP(011).

Команда STEP(008) должна располагаться в начале каждого из шагов. Эта команда располагается в начале области шага для указания начала шага.

**Конец области программирования шага**

Команда STEP(008), не содержащая операнда, располагается в конце области программирования шага для указания конца программирования шага. Когда бит управления, предшествующий команде SNXT(009), переводится в состояние ON, выполнение шага прекращается по команде SNXT(009).

**Флаги: STEP(008)**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда указанный бит В находится за пределами области WR. Переводится в состояние ON, когда команда STEP(008) используется в прерываемой программе. OFF в других случаях.

**Флаги: SNXT(009)**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда указанный бит В находится за пределами области WR. Переводится в состояние ON, когда команда SNXT(009) используется в прерываемой программе. OFF в других случаях.

**Меры предосторожности**

Бит управления В команд STEP(008)/SNXT(009) должен находиться в рабочей области.

Бит управления команд STEP(008)/SNXT(009) не может использоваться где-либо еще в релейно-контактной схеме. Если один и тот же бит используется дважды, определяется ошибка дублирования бита.

В случае, когда команда SBS(091) применяется в шаге для вызова подпрограммы из этого же шага, выводы подпрограммы и команды не будут блокироваться, когда бит управления переключается в состояние ON.

Биты управления в одном разделе шаговой программы должны быть последовательными битами одного слова.

Команда SNXT(009) будет выполняться только один раз, т.е. при возрастающем изменении условия выполнения.

Располагайте команду SNXT(009) в конце области программирования шага, убедившись в том, что бит управления является «холостым» битом, находящимся в рабочей области. Когда бит управления шага используется последней командой SNXT(009) в области программирования шага, выполнение соответствующего шага начинается при выполнении команды SNXT(009).

Когда операнд В, указанный для команд STEP(008)/SNXT(009), находится за пределами рабочей области, или когда шаговая программа располагается где-либо кроме циклической задачи, определяется ошибка и флаг ошибки переводится в состояние ON.

При выполнении команды STEP(008) флаг шага A20012 переводится в состояние ON на время одного цикла. Этот флаг может использоваться для проведения инициализации в начале выполнения шага.

**Условия расположения областей программирования шагов (STEP В в STEP)**

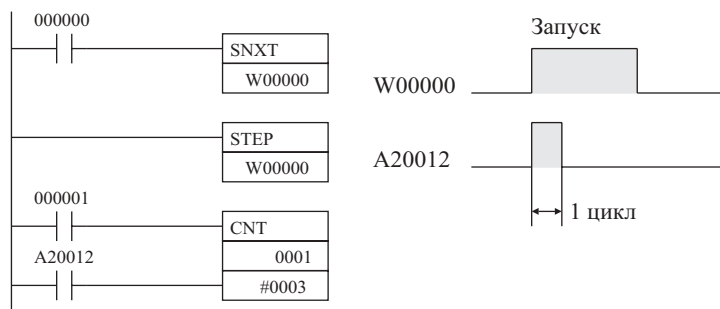
Команды STEP(008) и SNXT(009) не могут использоваться внутри подпрограмм, в программах прерывания или блочных программах.

Убедитесь в том, что в одном цикле задано выполнение не более одного шага.

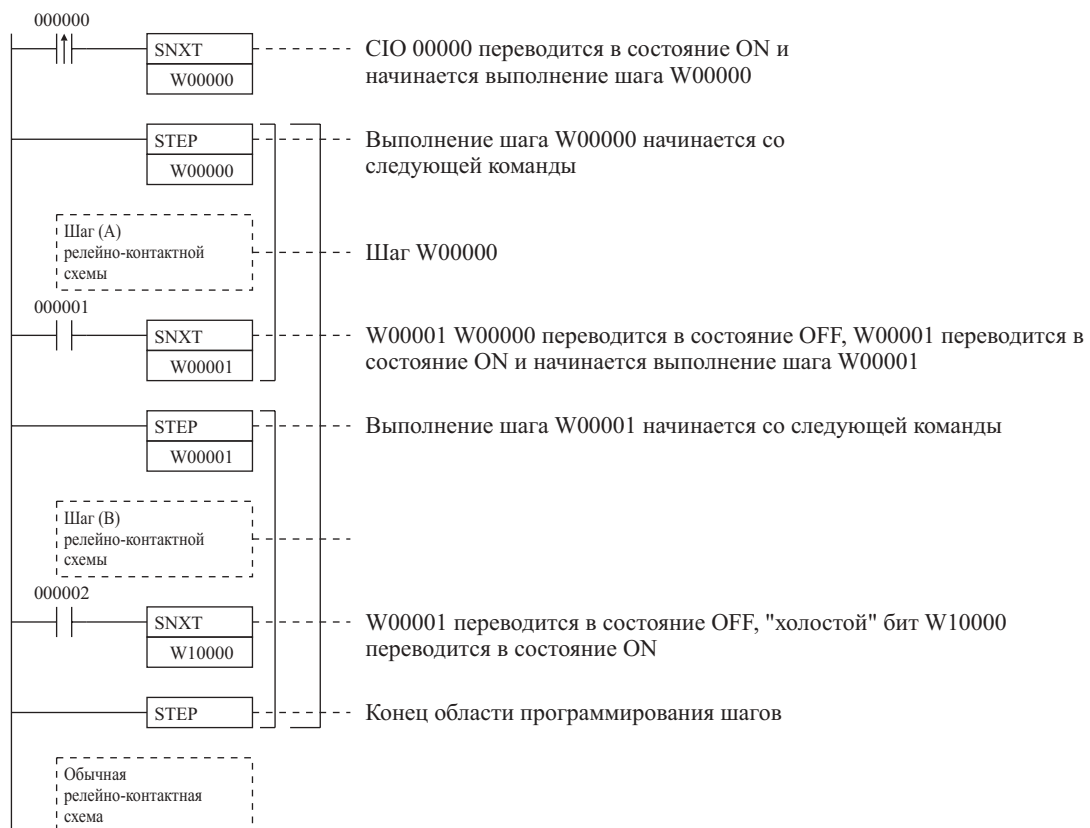
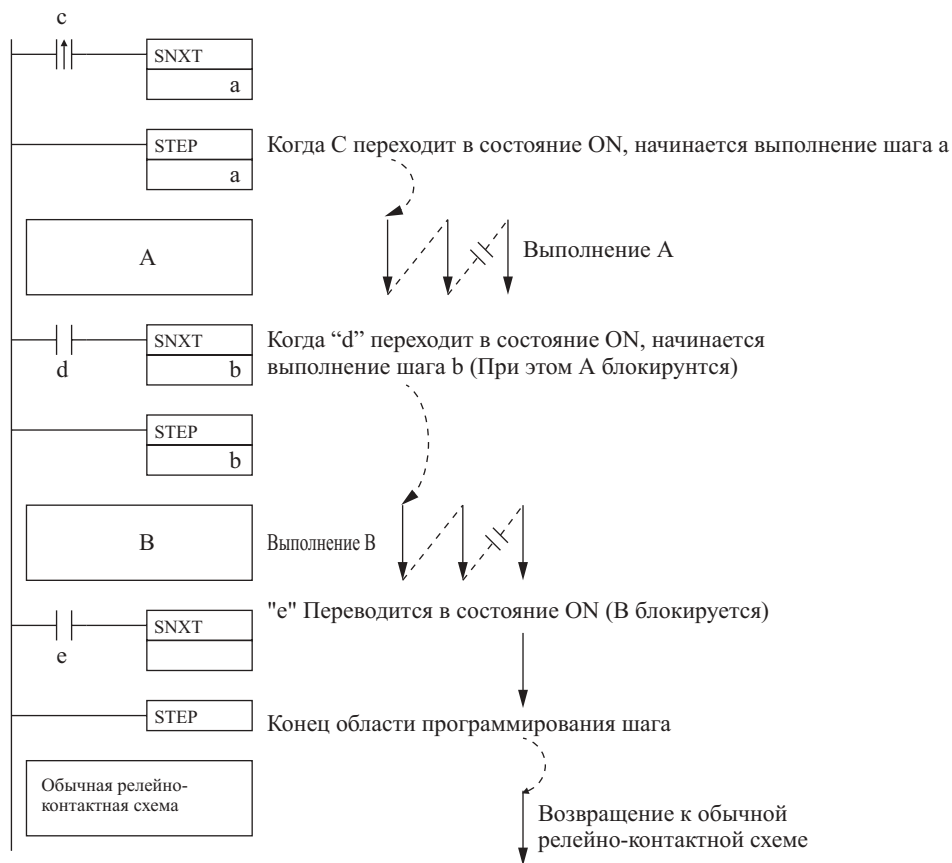
**Команды, применение которых в шаговых программах не допускается**

Команды, применение которых в шаговых программах не допускается, перечислены в следующем списке.

Функция	Мнемоническое обозначение	Наименование
Команды последовательного управления	END(001)	END
	IL(002)	INTERLOCK
	ILC(003)	INTERLOCK CLEAR
	JMP(004)	JUMP
	JME(005)	JUMP END
	CJP(510)	CONDITIONAL JUMP
	CJPN(511)	CONDITIONAL JUMP NOT
	JMP0(515)	MULTIPLE JUMP
JME0(516)	MULTIPLE JUMP END	
Команды подпрограмм	SBN(092)	SUBROUTINE ENTRY
	RET(093)	SUBROUTINE RETURN

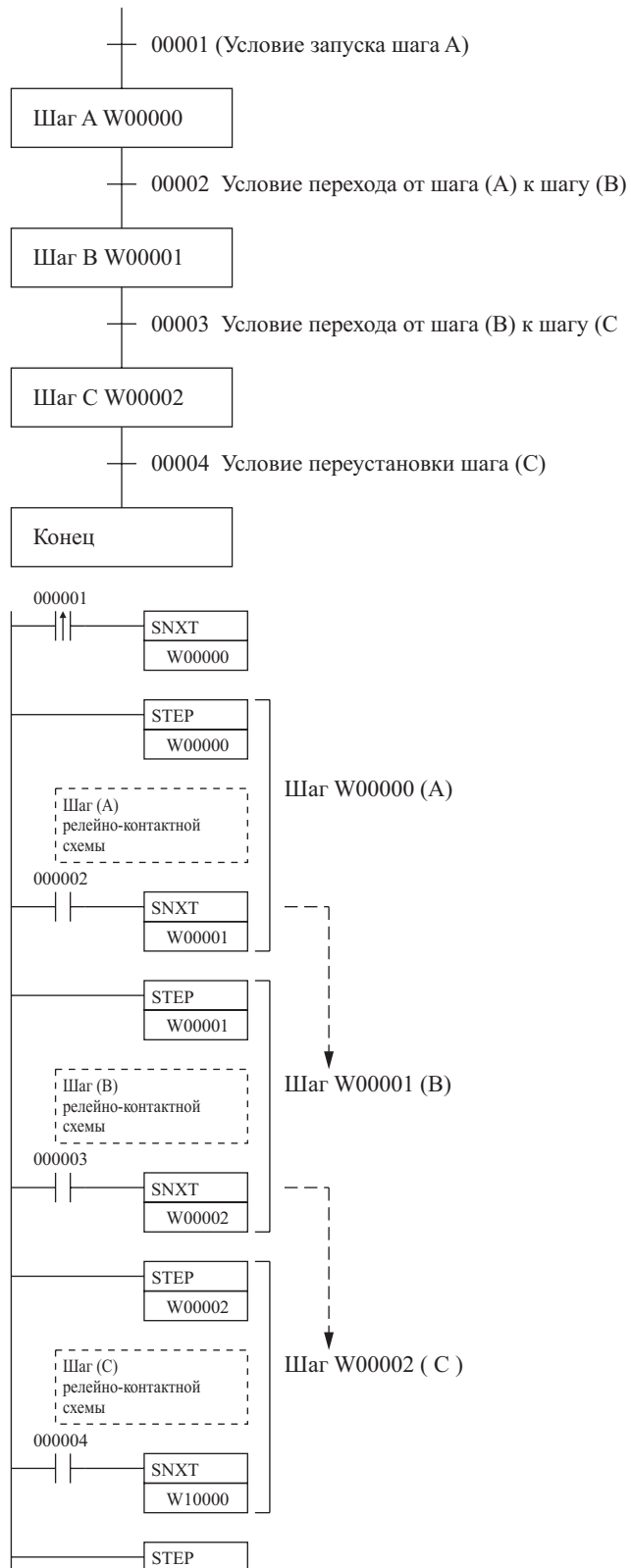
**Биты, относящиеся к команде**

Наименование	Адрес	Подробности
Флаг шага	A20012	Переводится в состояние ON на время выполнения одного цикла, когда по команде STEP(008) начинается выполнение шаговой программы. Может использоваться для переустановки таймеров и выполнения других операций при запуске выполнения нового шага.

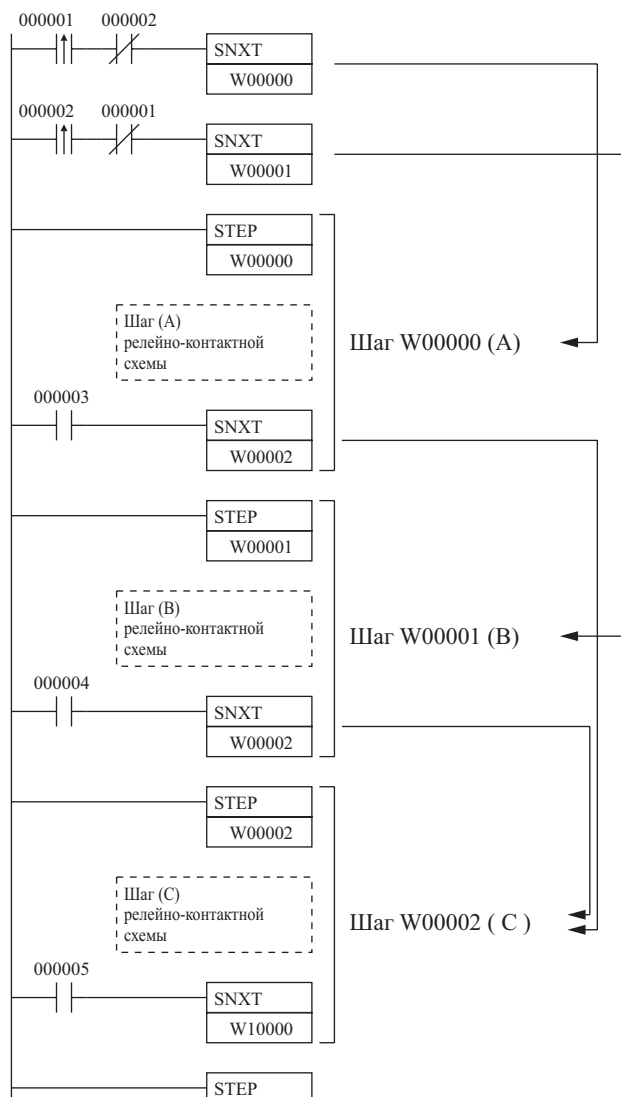
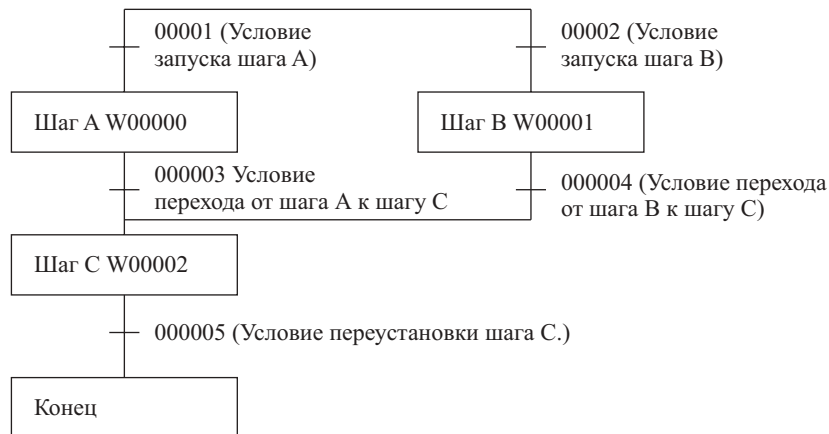


## Примеры

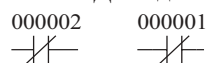
## Последовательное управление



## Управление с разветвлением

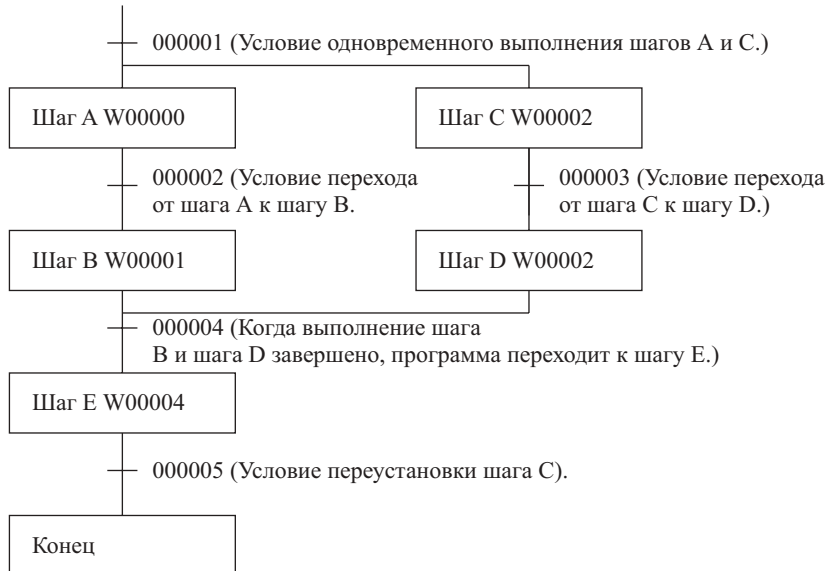


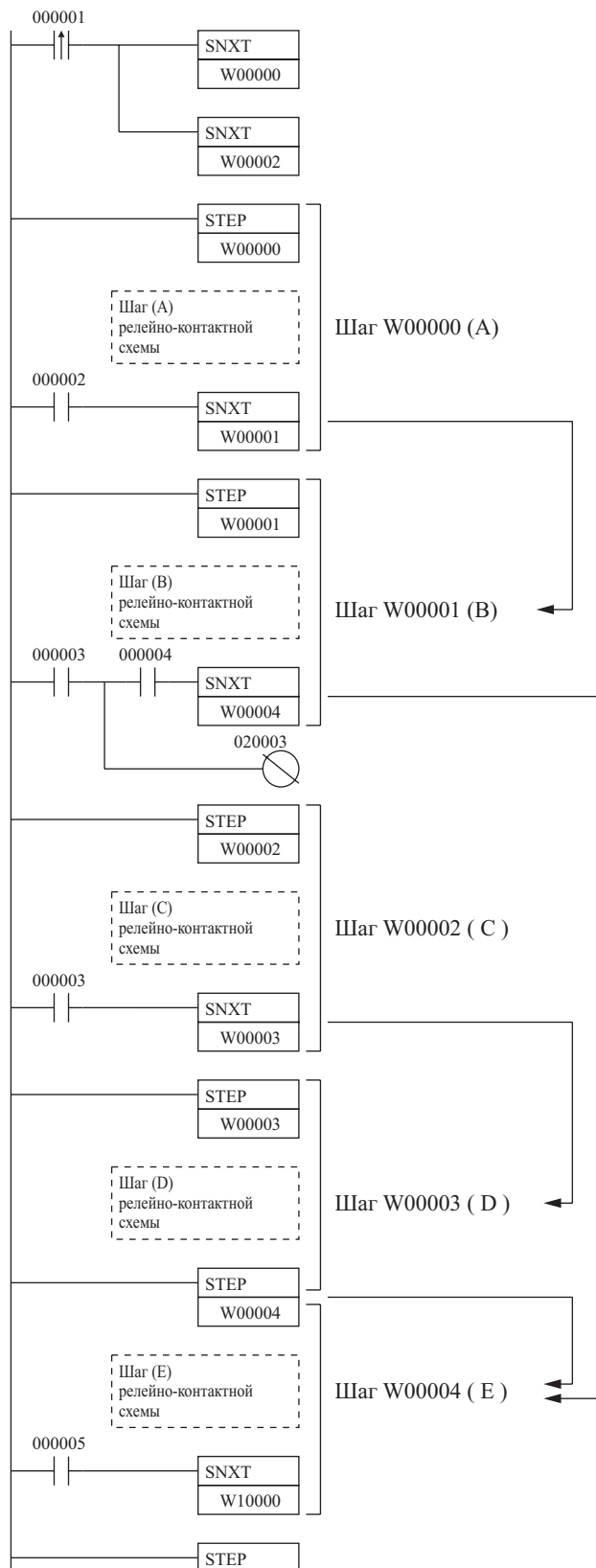
Приведенные выше примеры программ используются, когда шаги А и В не могут выполняться одновременно. Для одновременного выполнения А и В удалите условия выполнения, показанные ниже.



**Примечание:** В приведенном выше примере, когда для W00002 выполняется команда SNXT(009), ответвление переходит к следующему шагу, даже если один и тот же бит используется дважды. При проверке программы с помощью СХ- программатора такая ситуация не принимается в качестве ошибки. Ошибка дублирования бита определяется только тогда, когда бит управления, используемый в шаговых командах, используется также в обычной релейно-контактной программе.

#### Параллельное управление





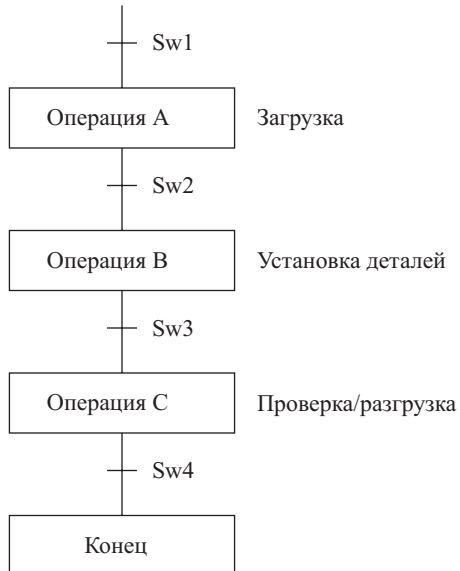
### Примеры применения

Следующие ниже три примера демонстрируют три типа управления, возможных при использовании шаговой программы. Пример 1 демонстрирует последовательное выполнение операций, пример 2 – последовательно-параллельное выполнение операций, пример 3 – параллельное выполнение операций.

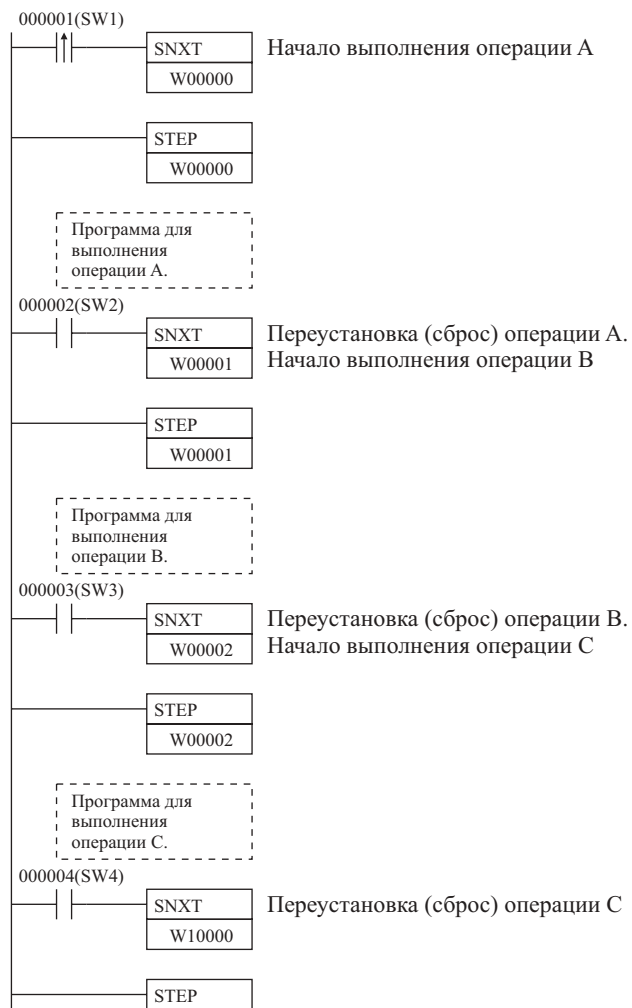


**Пример 1: Последовательные действия**

Приводимый ниже процесс предполагает, что три операции, включая загрузку, установку деталей и проверку/разгрузку, выполняются последовательно, производя установку каждой операции в исходное состояние (сброс) перед выполнением следующей операции. Различные датчики (SW1, SW2, SW3 и SW4) располагаются таким образом, чтобы подавать сигнал о начале и завершении каждой из операций. На следующем ниже рисунке представлен ход выполнения операций и переключения, выполняемые для управления процессом.



Программа, созданная для выполнения такого процесса, использует базовые элементы программирования шагов: каждый из шагов заканчивается отдельной командой SNXT(009), которая запускает выполнение следующего шага. Выполнение каждого из шагов начинается после того, как переключатель, обозначающий окончание выполнения предшествующего шага, переводится в состояние ON.

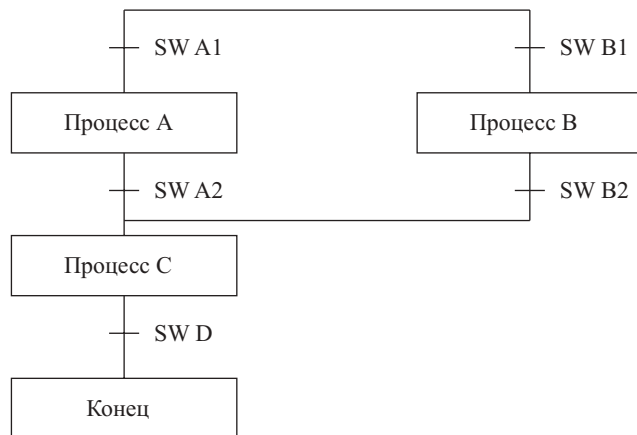


Адрес	Команда	Операнды
000000	@LD	000001
000001	SNXT(009)	W 00000
000002	STEP(008)	W 00000
Операция А		
000100	LD	000002
000101	SNXT(009)	W 00001
000102	STEP(008)	W 00001
Операция В		
000100	LD	000003
000101	SNXT(009)	W 00002
000102	STEP(008)	W 00002
Операция С		
000200	LD	000004
000201	SNXT(009)	W 00003
000202	STEP(008)	W 00003

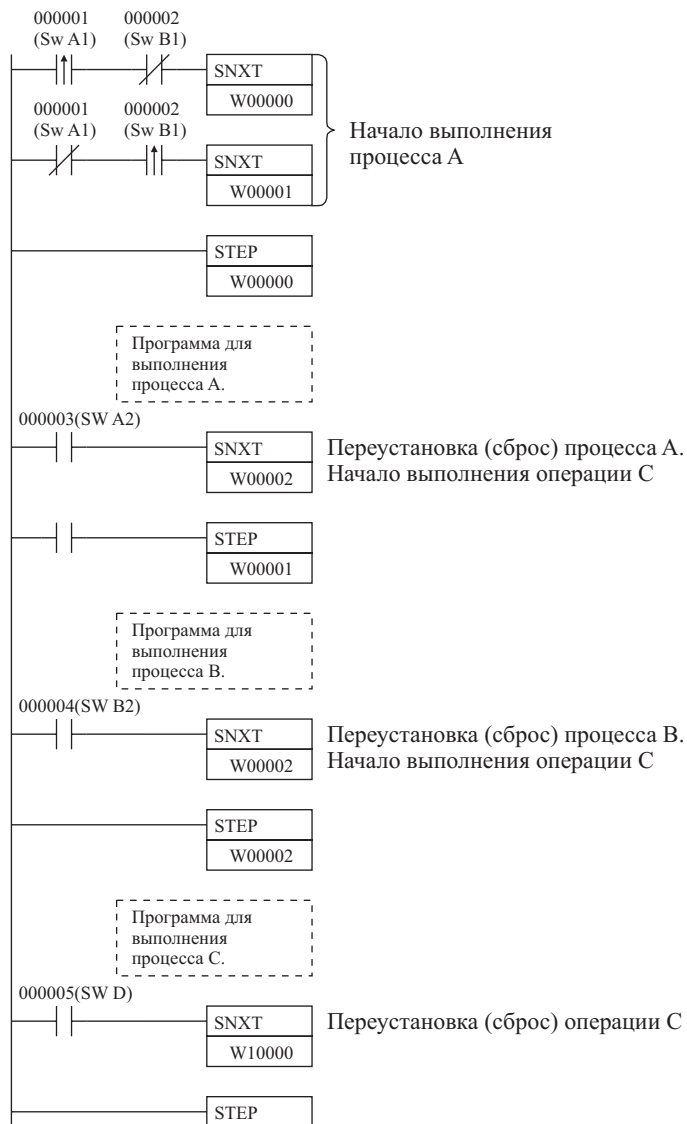
### Пример 2: Последовательно-параллельное выполнение операций

В следующем ниже примере обработка продукции перед маркировкой выполняется в одном из двух направлений, в зависимости от веса обрабатываемой продукции. Процесс маркировки одинаков, вне зависимости от процесса предварительной обработки. Различные датчики установлены для подачи сигналов, после которых начинается или заканчивается выполнение операции.

На следующем ниже рисунке представлен ход выполнения операций и переключения, выполняемые для управления процессом. В данном случае в зависимости от состояния переключателей SW A1 и SW B1 выполняется процесс обработки А или процесс обработки В.



Программа, созданная для выполнения такого процесса, начинается двумя командами SNXT(009), которые запускают выполнение процесса обработки А и процесса обработки В. Используемый способ программирования для CIO 000001 (SW A1) и CIO 000002 (SW B1) предполагает срабатывание одного из переключателей с установлением условия выполнения (ON) для выполнения процесса обработки А или процесса обработки В. Оба процесса обработки заканчиваются командой SNXT(009), которая запускает выполнение операции С.



Адрес	Команда	Операнды
000000	@LD	000001
000001	AND NOT	000002
000002	SNXT(009)	010000
000003	LD NOT	000001
000004	@AND	000002
000005	SNXT(009)	010001
000006	STEP(008)	010000
Процесс А		
000100	LD	000003
000101	SNXT(009)	010002
000102	STEP(008)	010001
Процесс В		
000100	LD	000004
000101	SNXT(009)	010002
000102	STEP(008)	010002
Процесс С		
000200	LD	000005

Адрес	Команда	Операнды
000201	SNXT(009)	024614
000202	STEP(008)	–

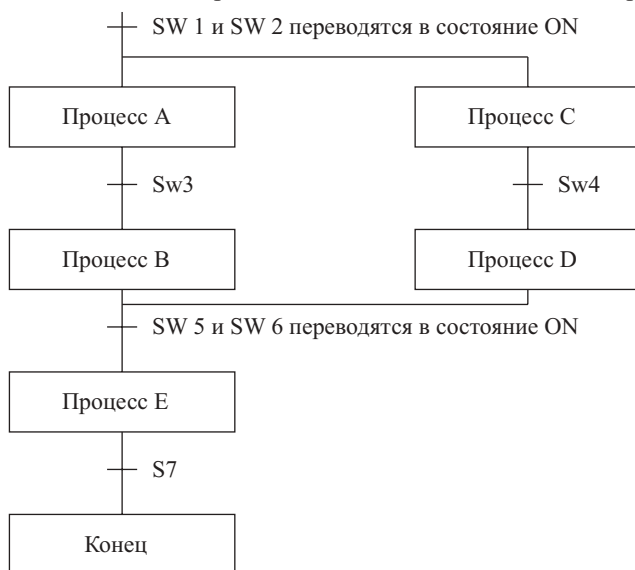
**Примечание:** В описанной выше программе СЮ 010002 используется в двух командах SNXT(009). Это не приводит к определению ошибки дублирования в процессе проверки программы.

### Пример 3: Параллельное выполнение операций

В следующем ниже примере две детали проходят одновременную обработку в двух процессах, затем они объединяются в следующем процессе.

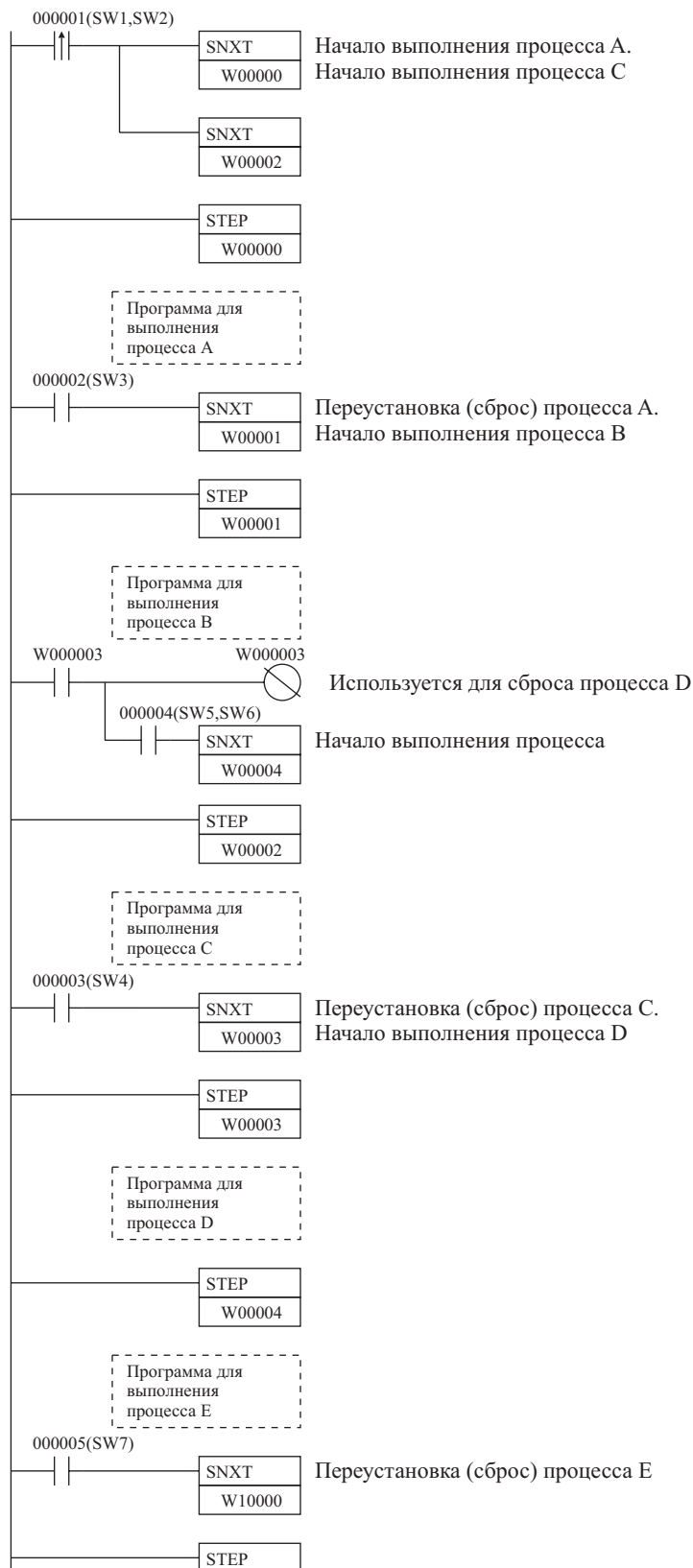
Различные датчики установлены для подачи сигналов, которые сигнализируют о начале или завершении выполнения операций.

На следующем рисунке представлен ход выполнения операций и переключения, используемые для управления процессом. В данном случае процесс обработки А и процесс обработки С начинаются одновременно. После завершения выполнения процесса А начинается выполнение процесса В, после завершения выполнения процесса С начинается выполнение процесса D.



Программа, созданная для выполнения такого процесса, начинается двумя командами SNXT(009), которые запускают выполнение процесса обработки А и процесса обработки С. Эти команды разветвляются из одной линии команд и всегда выполняются одновременно, запуская выполнение шагов для процесса А и процесса С. После выполнения процессов А и С, начинается выполнение шагов в процессах В и D.

После окончания выполнения процесса В и процесса D (т.е. когда SW5 и SW6 переходят в состояние ON), по команде SNXT(009) производится их переустановка в конце программы для процесса В. Несмотря на то, что в конце процесса D отсутствует команда SNXT(009), бит управления для этого процесса переводится в состояние OFF посредством выполнения команды SNXT(009) W00004. Это происходит потому, что команда OUT для бита W00003 находится в шаге, сбрасываемом командой SNXT(009) W00004, т.е. W00003 переводится в состояние OFF, когда выполняется команда SNXT(009) W00004. Таким образом, процесс В переустанавливается напрямую, а процесс D переустанавливается косвенно перед выполнением процесса E.



Адрес	Команда	Операнды
000000	@LD	000001
000001	SNXT(009)	W 00000

---

Адрес	Команда	Операнды
000002	SNXT(009)	W 00002
000003	STEP(008)	W 00000
Процесс А		
000100	LD	000002
000101	SNXT(009)	W 00001
000102	STEP(008)	W 00001
Процесс В		
000100	LD	000003
000101	OUT	W 00003
000101	AND	000004
000101	SNXT(009)	W 00004
000102	STEP(008)	W 00002
Процесс С		
000200	LD	000003
000201	SNXT(009)	W 00003
000202	STEP(008)	W 00003
Процесс D		
000300	STEP(008)	W 00004
Процесс E		
000400	LD	000005
000401	SNXT(009)	024613
000402	STEP(008)	–

---

## 3-21 Команды для базовых модулей ввода/вывода

Настоящий раздел описывает команды, используемые для управления Модулями ввода/вывода.

Команда	Мнемоническое изображение	Функциональный код	Страница
I/O REFRESH	IORF	097	569
7-SEGMENT DECODER	SDEC	078	571
INTELLIGENT I/O READ	IORR	222	574
INTELLIGENT I/O WRITE	IOWR	223	577

## 3-21-1 Команда регенерации ввода/вывода I/O REFRESH: IORF(097)

**Назначение**

По команде IORF(097) производится регенерация указанных слов ввода/вывода.

**Символ релейно-контактной схемы**

—	IORF(097)	
	St	St: Начальное слово
	E	E: Последнее слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	IORF(097)
	Выполняется один раз для дифференцирования вверх	@IORF(097)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****St: Начальное слово**

От CIO 0000 до CIO 0999 (Область битов ввода/вывода), или  
От CIO 2000 до CIO 2959 (Область битов Специальных модулей).

**E: Последнее слово.**

От CIO 0000 до CIO 0999 (Область битов ввода/вывода), или  
От CIO 2000 до CIO 2959 (Область битов Специальных модулей).

**Примечание:** St и E должны находиться в одной области памяти.

**Спецификации операндов**

Область	St	E
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	—	
Область удержания бита (Holding bit area)	—	
Область вспомогательных битов	—	
Область таймера	—	
Область счетчика	—	
Область DM	—	
Область EM, не содержащая банков	—	
Область EM, содержащая банки	—	
Адреса DM/EM памяти при косвенном указании в двоичном коде	—	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—	
Константы	—	
Регистры данных	—	
Индексные регистры	—	



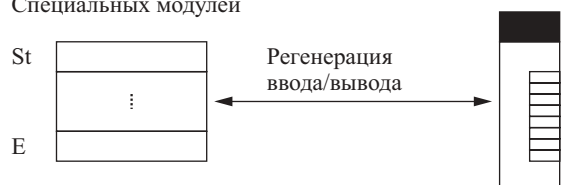
Область	St	E
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0...-2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...IR15(++) ,(- -)IR0...(- -)IR15	

### Описание

По команде IORF(097) производится регенерация слов ввода/вывода между St и E, включая St и E. Команда IORF(097) применяется для регенерации слов ввода/вывода, распределенных Базовым модулям ввода/вывода или Специальным модулям, установленным в Панель Центрального процессора или Панели расширения. Команда IOFR(097) не может применяться для регенерации слов одновременно в обеих областях (т.е. с помощью одной команды). Базовым модулям ввода вывода распределяются слова между CIO 0000 и CIO 0999, а Специальным модулям распределяются слова от CIO 2000 до CIO 2959.

Когда задается выполнение регенерации слов в области битов Специальных модулей, все 10 слов, распределенных Модулю, будут подвергнуты регенерации, если первое слово из десяти будет включено в указываемый перечень слов.

Область битов ввода/вывода или область битов Специальных модулей



Если в диапазоне слов между St и E существуют слова, для которых не существует соответствующего установленного Модуля, с этими словами ничего не происходит, т.е. регенерация производится только для слов, распределенных соответствующим Модулем.

Как Специальные модули C200H, так и Специальные модули серии CS, могут подвергаться регенерации с помощью одной и той же команды.

Все слова, распределенные Высокоскоростным модулям ввода/вывода C200H группа 2, должны подвергаться регенерации за один раз. Слова ввода/вывода Модуля будут подвергнуты регенерации, если первое слово, распределенное Модулю, указывается в перечне слов ввода/вывода для выполнения регенерации. (Слова, распределенные Модулю, не будут подвергнуты регенерации, если начальное слово находится после первого из слов, распределенных модулю, однако они будут подвергнуты регенерации, даже если задаваемое последнее слово находится перед последним словом, распределенным Модулю.)

Команда IOFR(097) может использоваться в задачах прерывания, обеспечивая высокое быстродействие регенерации отдельных слов ввода/вывода. (См. меры предосторожности.)

### Применяемые Модули

С помощью команды IORF(097) можно производить регенерацию указанных ниже модулей. Эти модули могут подвергаться регенерации только тогда, когда они установлены в Панель центрального процессора или в Панель расширения. Регенерация этих Модулей не может выполняться, если они установлены в Slave – панели.

Базовые модули CS1, Базовые модули C200H, Модули группы 2, и специальные модули (включая высокоскоростные модули). (Все слова, распределенные модулям, могут подвергаться регенерации.)

**Примечание:** Модули, которые подвергаются регенерации по команде IORF(097), не обязательно являются модулями, регенерация которых может производиться по указанию немедленной регенерации (!).

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	<p>Переводится в состояние ON, когда St больше E.</p> <p>Переводится в состояние ON, когда адрес St при косвенном указании его с помощью индексного регистра находится за допустимыми пределами.</p> <p>Переводится в состояние ON, когда адрес E при косвенном указании его с помощью индексного регистра находится за допустимыми пределами.</p> <p>Переводится в состояние ON, когда St и E находятся в различных областях памяти.</p> <p>OFF в других случаях.</p>

**Меры предосторожности**

В случае, когда слова области битов ввода/вывода (CIO 0000...CIO 0999) и области битов Специальных модулей (CIO 2000...CIO 2959) указываются в одной команде, определяется ошибка.

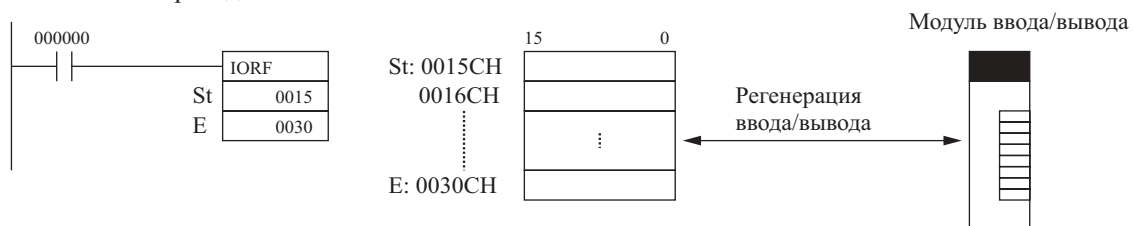
Регенерация ввода/вывода Модулей не будет выполняться, если для этих Модулей определена ошибка таблицы ввода/вывода.

Регенерация ввода/вывода, выполняемая по команде IORF(097), прерывается, если в процессе выполнения возникает ошибка шины ввода/вывода.

Когда команда IORF(097) используется в задаче прерывания, непременно отключите в начальных установках Программируемого контроллера выполнение периодической регенерации Специальных модулей. Если такая операция не производится, при выполнении периодической регенерации и регенерации по команде IORF определяется допускаемая ошибка двойной регенерации и флаг ошибки задачи прерывания (A40213) переводится в состояние ON.

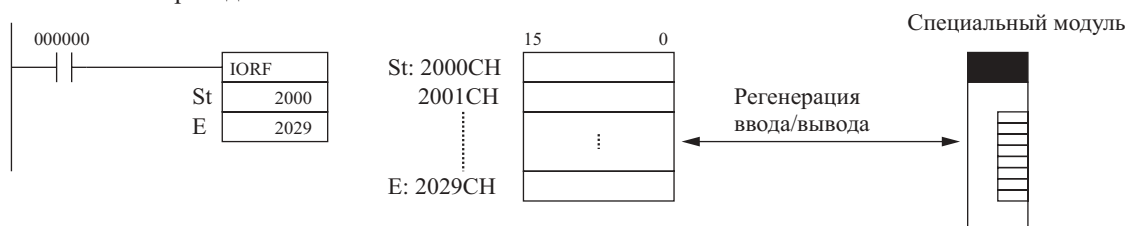
**Примеры****Регенерация слов в области битов ввода/вывода**

В следующем ниже примере показано выполнение регенерации 16-ти слов от CIO 0015 до CIO 0030, когда CIO 000000 переводится в состояние ON.

**Регенерация ввода/вывода. Модуль ввода/вывода.**

Регенерация слов в области битов Специальных модулей ввода/вывода

В следующем ниже примере показано выполнение регенерации 30-ти слов от CIO 2000 до CIO 2029, когда CIO 000000 переводится в состояние ON.

**3-21-2 Команда преобразования в код для семисегментного дисплея 7-SEGMENT DECODER: SDEC(078)****Назначение**

По команде SDEC(078) производится преобразование шестнадцатеричного содержания указанной цифры (цифр) в код семисегментного дисплея размером 8 битов и вывод этого кода в старшие или младшие 8 битов указанных слов назначения.

## Символ релейно-контактной схемы

SDEC(078)	
S	S: Исходное слово
Di	Di: Указатель цифры
D	D: Первое слово назначения

## Модификации

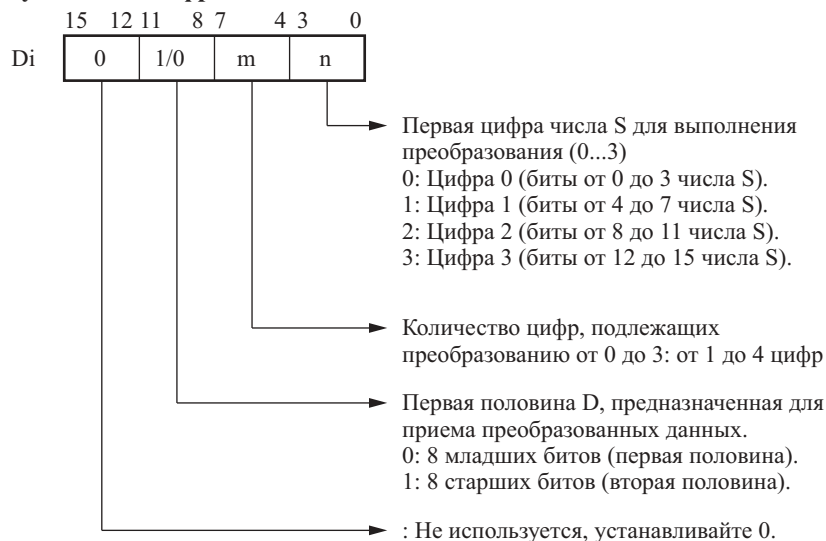
Модификации	Выполняется в каждом цикле для состояния ON	SDEC(078)
	Выполняется один раз для дифференцирования вверх	@ SDEC(078)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

## Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

## Операнды

Di: указатель цифры.



## Спецификации операндов

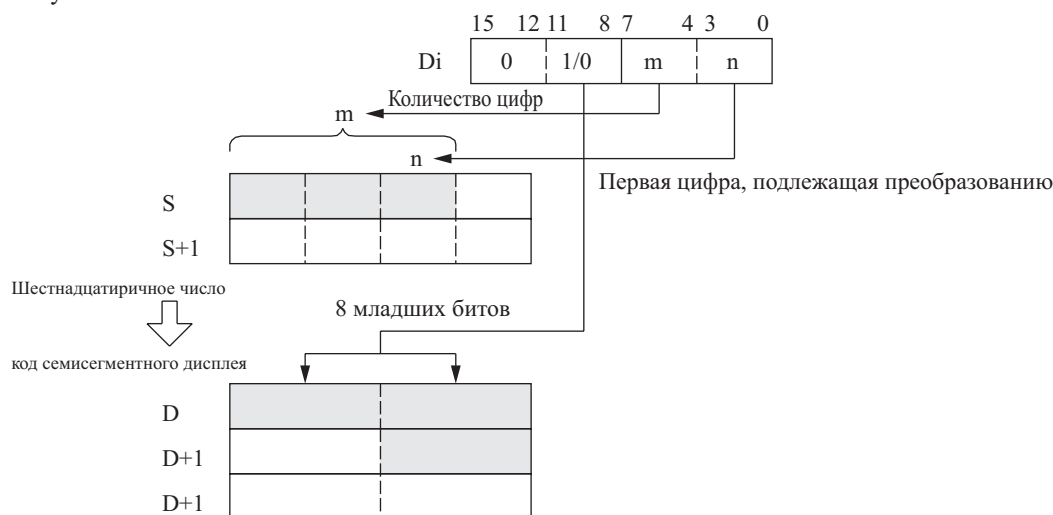
Область	S	Di	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A959		A448...A959
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы		Только указанные значения	–

Область	S	Di	D
Регистры данных	DR0...DR15		—
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0...-2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,(- -)IR0...,(- -)IR15		

**Описание**

Команда SDEC(078) принимает данные, указанные в S, как четырехзначные шестнадцатеричные данные, преобразует цифры, указанные в Di (первая цифра и количество цифр) в код семисегментного дисплея и выводит результат в слово назначения D, согласно указанным в Di битам слова D.

Рисунок.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда установки в Di выходят за допустимые пределы. OFF в других случаях.

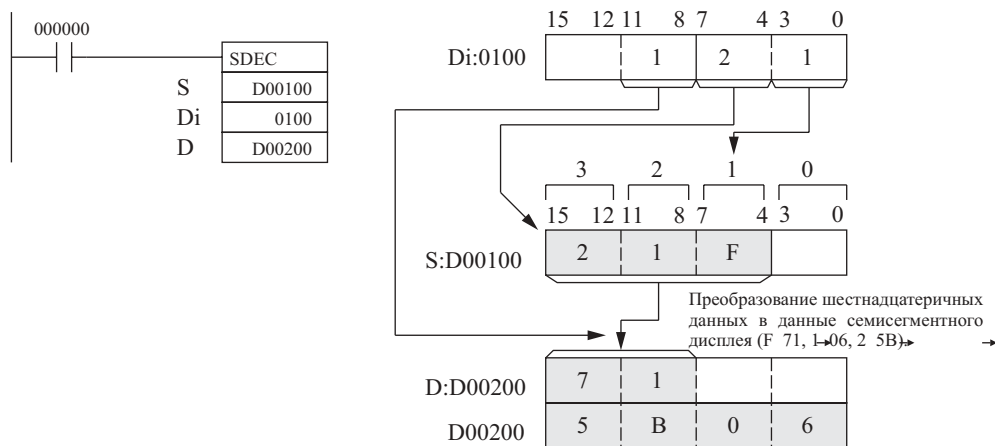
**Меры предосторожности**

Когда в Di для преобразования указывается более одной цифры, преобразование цифр производится по порядку, начиная с наименее значащей цифры. При этом цифра 0 следует за цифрой 3.

Результат записывается в D по порядку, начиная с указанной части и по направлению к словам с большими адресами. Если только один из байтов слова назначения принимает преобразованные данные, другой оставляется без изменений.

**Примеры**

В следующем ниже примере, когда CIO 000000 переводится в состояние ON, содержание трех цифр, начиная с цифры 1 в D00100, преобразуется из шестнадцатеричных данных в данные для семисегментного дисплея. Результат выводится в старший байт слова D00200 и оба байта слова D00201. Спецификация байтов и расположение выходных байтов указывается в CIO 0100.



**Данные для семисегментного дисплея**

Следующая ниже таблица показывает преобразование шестнадцатеричных чисел (4 бита) в код семисегментного дисплея (8 битов).

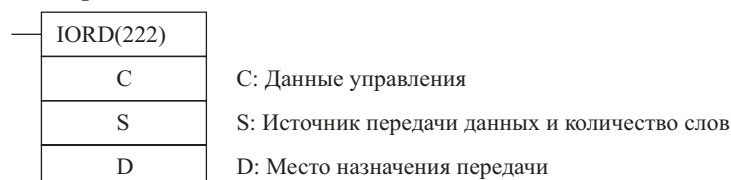
		Исходные данные												
		.....											Шестн.	
Исходные данные		Преобразование в код (сегменты)											Дисплей	
Цифра	Биты	-	g	f	e	d	c	b	a	Hex				
0	0 0 0 0	0	0	1	1	1	1	1	1	3F	0			
1	0 0 0 1	0	0	0	0	0	1	1	0	06	1			
2	0 0 1 0	0	1	0	1	1	0	1	1	5B	2			
3	0 0 1 1	0	1	0	0	1	1	1	1	4F	3			
4	0 1 0 0	0	1	1	0	0	1	1	0	66	4			
5	0 1 0 1	0	1	1	0	1	1	0	1	6D	5			
6	0 1 1 0	0	1	1	1	1	1	1	0	7D	6			
7	0 1 1 1	0	0	1	0	0	1	1	1	27	7			
8	1 0 0 0	0	1	1	1	1	1	1	1	7F	8			
9	1 0 0 1	0	1	1	0	1	1	1	1	6F	9			
A	1 0 1 0	0	1	1	1	0	1	1	1	77	A			
B	1 0 1 1	0	1	1	1	1	1	0	0	7C	b			
C	1 1 0 0	0	0	1	1	1	0	0	1	39	C			
D	1 1 0 1	0	1	0	1	1	1	1	0	5E	d			
E	1 1 1 0	0	1	1	1	1	0	0	1	79	E			
F	1 1 1 1	0	1	1	1	0	0	0	1	71	F			

**3-21-3 Команда чтения памяти ввода/вывода INTELLIGENT I/O READ: IORD(222)**

**Назначение**

По команде IORD(222) производится чтение памяти Модуля ввода/вывода.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	IORD(222)
	Выполняется один раз для дифференцирования вверх	@ IORD(222)
	Выполняется один раз для дифференцирования вниз	Не поддерживается

Указание немедленной ре-генерации	Не поддерживается
-----------------------------------	-------------------

### Применяемые области программы

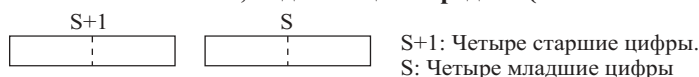
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

### Операнды

**C:** Изменяется в зависимости от используемого Специального модуля.

**S:** Номер модуля от 0000 до 005F (0...95).

**S+1:** Количество слов, подлежащих передаче (зависит от типа Специального модуля).



### Спецификации операндов

Область	C	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6142	CIO 0000...CIO 6143
Рабочая область	W000...W511	W000...W510	W000...W511
Область удержания бита (Holding bit area)	H000...H511	H000...H510	H000...H511
Область вспомогательных битов	A000...A959	A000...A958	A448...A959
Область таймера	T0000...T4095	T0000...T4094	T0000...T4095
Область счетчика	C0000...C4095	C0000...C4094	C0000...C4095
Область DM	D00000...D32767	D00000...D32766	D00000...D32767
Область EM, не содержащая банков	E00000...E32767	E00000...E32766	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32766 (n = 0...C)	En_00000...En_32767 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF (двоичные)	Только указанные значения	—
Регистры данных	DR0...DR15		
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0...-2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15		

### Описание

По команде IORD(222) производится чтение заданного в S+1 количества слов в памяти Специальных модулей, чьи номера обозначены в слове S, и вывод данных в слово D. Могут указываться только модули, установленные в Панели центрального процессора или Панели расширения ввода/вывода. Для детального ознакомления с данными, которые можно читать в каждом из модулей, обратитесь к Руководству по эксплуатации соответствующего Специального модуля.



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда количество слов, подлежащих передаче (S), выходит за пределы от 0001 до 0080 (шестн.) Переводится в состояние ON, когда номер модуля (S), выходит за пределы от 0001 до 005F (шестн.) Переводится в состояние ON, когда модуль назначения работает в качестве модуля SYSMAC BUS. Переводится в состояние ON, когда указывается Специальный модуль, на который команда IORD(222) не воздействует. Переводится в состояние ON, когда указывается Специальный модуль, в котором определяется ошибка установки или ошибка функционирования. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда операция чтения завершается успешно. Переводится в состояние OFF, когда операция чтения не завершается успешно.

**Меры предосторожности**

Флаг равенства переводится в состояние ON, когда операция чтения завершается успешно.

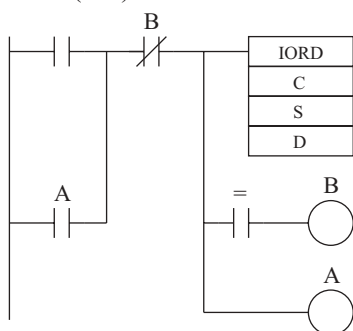
Флаг равенства переводится в состояние OFF, когда операция чтения не завершается успешно вследствие того, что Специальный модуль занят.

При возникновении одной из перечисленных ниже ошибок определяется ошибка, и флаг ошибки переводится в состояние ON.

- Когда количество слов, подлежащих передаче (S), выходит за пределы от 0001 до 0080 (шестн.).
- Когда номер модуля (S), выходит за пределы от 0001 до 005F (шестн.).
- Когда модуль назначения работает в качестве модуля SYSMAC BUS.
- Когда осуществляется доступ к Специальному модулю, на который команда IORD(222) не воздействует.
- Когда осуществляется доступ к Специальному модулю, в котором определяется ошибка установки или ошибка функционирования.

При выполнении команды IORD(222) результаты отражаются на состоянии флагов состояния. В частности, флаг равенства переводится в состояние ON, когда чтение завершается успешно. Вводите флаги состояния, такие как флаги равенства, с разветвлением на выходе в качестве условия выполнения команды IORD(222) в следующем цикле.

В случае, когда Специальный модуль оказывается занятым, операция чтения не выполняется. Используйте Флаг равенства для создания удерживающейся программы, как показано ниже, так, чтобы команда IORD(222) выполнялась в каждом из циклов до выполнения операции чтения.

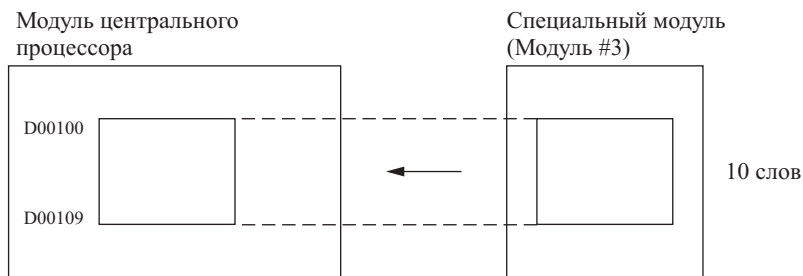
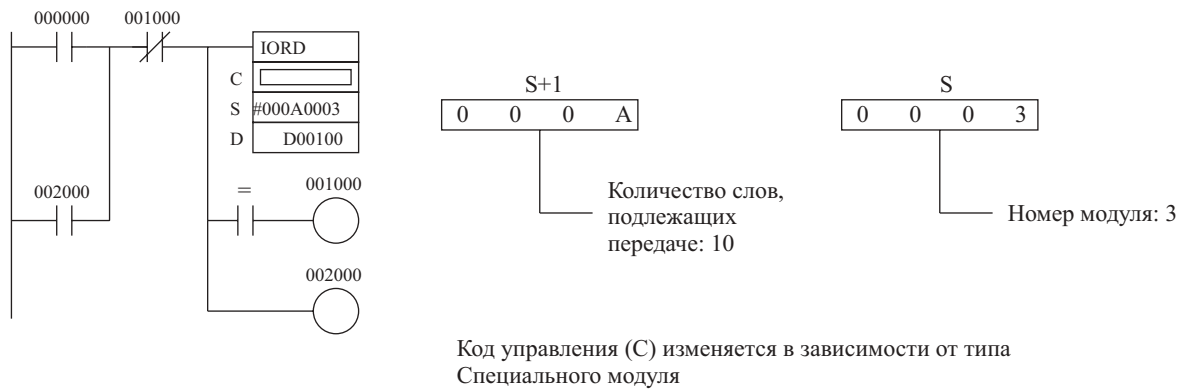


Когда возникают условия ввода, условие поддержания обеспечивается выводом А и командой IORD(222), выполняемой в каждом из циклов до тех пор, пока флаг равенства не будет переведен в состояние ON. После завершения чтения и перевода флага равенства в состояние ON, вывод В переводится в состояние ON и условие поддержания сбрасывается.

Непосредственно после команд IORD(222) непременно помещайте флаги состояния, и не вводите эти флаги после любой другой команды. Если флаг состояния располагается после другой команды, он будет подвержен воздействию результата выполнения этой команды.

**Пример**

В данном примере команда IORD(222) используется для чтения данных.



### 3-21-4 Команда записи в память ввода/вывода INTELLIGENT I/O READ: IOWR(223)

#### Назначение

По команде IOWR(223) производится вывод содержания памяти ввода/вывода из Модуля центрального процессора в Специальный модуль.

#### Символ релейно-контактной схемы

IOWR(223)	
C	C: Данные управления
S	S: Источник передачи данных и количество слов
D	D: Место назначения передачи и количество слов

#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	IOWR(223)
	Выполняется один раз для дифференцирования вверх	@ IOWR(223)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

**C:** Изменяется в зависимости от используемого Специального модуля.

**D:** Номер модуля от 0000 до 005F (0...95).

**D+1:** Количество слов, подлежащих передаче (зависит от типа Специального модуля).



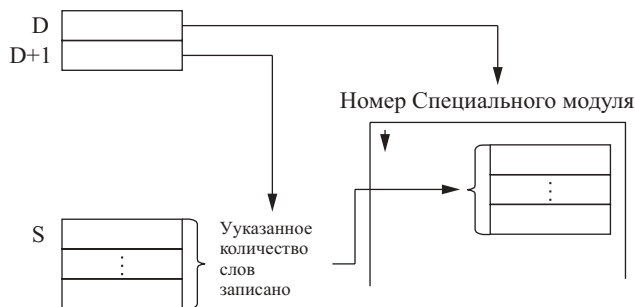


### Спецификации операндов

Область	C	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6142	
Рабочая область	W000...W511	W000...W510	
Область удержания бита (Holding bit area)	H000...H511	H000...H510	
Область вспомогательных битов	A000...A959	A000...A958	
Область таймера	T0000...T4095	T0000...T4094	
Область счетчика	C0000...C4095	C0000...C4094	
Область DM	D00000...D32767	D00000...D32766	
Область EM, не содержащая банков	E00000...E32767	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	#0000...#FFFF (двоичные)	Только указанные значения	
Регистры данных	DR0...DR15	–	–
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048 +2047,IR0...-2048...+2047,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...IR15(++) ,-(-)IR0...,-(-)IR15		

### Описание

По команде IOWR(222) производится запись указанного в D количества слов, начиная с первого слова источника (указанного в S) и далее, и вывод этих слов в Специальный модуль, имеющий номер, указанный в D. В качестве указываемых модулей могут быть только Специальные модули, установленные в Панели Модуля центрального процессора или в Панели расширения ввода/вывода.



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	<p>Переводится в состояние ON, когда количество слов, подлежащих передаче (D), выходит за пределы от 0001 до 0080 (шестн.)</p> <p>Переводится в состояние ON, когда номер модуля (D), выходит за пределы от 0001 до 005F (шестн.)</p> <p>Переводится в состояние ON, когда D указывается константой, а количество слов, подлежащих передаче (D+1) не равно 0001 (шестн.)</p> <p>Переводится в состояние ON, когда модуль назначения работает в качестве модуля SYSMAC BUS.</p> <p>Переводится в состояние ON, когда указывается Специальный модуль, на который команда IOWR(223) не воздействует.</p> <p>Переводится в состояние ON, когда указывается Специальный модуль, в котором определяется ошибка установки или ошибка функционирования.</p> <p>OFF в других случаях.</p>

Наименование	Обозначение	Функционирование
Флаг равенства	=	Переводится в состояние ON, когда операция записи завершается успешно. Переводится в состояние OFF, когда операция записи успешно не завершается.

### Меры предосторожности

Когда в качестве количества слов, подлежащих передаче (D+1) указывается «0001», данные в D могут указываться константой. Если D указывается константой, а количество слов, подлежащих передаче, не равно «0001», определяется ошибка и флаг ошибки переводится в состояние ON.

Флаг равенства переводится в состояние ON, когда операция записи завершается успешно.

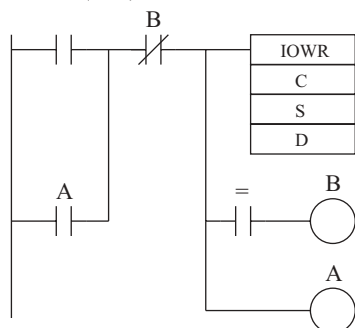
Флаг равенства переводится в состояние OFF, когда операция записи не завершается успешно вследствие того, что Специальный модуль занят.

При возникновении одной из перечисленных ниже ошибок определяется ошибка, и флаг ошибки переводится в состояние ON.

- Когда определяется ошибка проверки Модуля ввода/вывода, ошибка установки Специального модуля, ошибка функционирования Специального модуля.
- Когда количество слов, подлежащих передаче (D), выходит за пределы от 0001 до 0080 (шестн.).
- Когда номер модуля (D), выходит за пределы от 0001 до 005F (шестн.).
- Когда модуль назначения работает в качестве модуля SYSMAC BUS.
- Когда осуществляется доступ к Специальному модулю, на который команда IOWR(223) не воздействует.
- Когда осуществляется доступ к Специальному модулю, в котором определяется ошибка установки или ошибка функционирования.

При выполнении команды IOWR(223) результаты отражаются на состоянии флагов состояния. В частности, флаг равенства переводится в состояние ON, когда запись завершается успешно. Вводите флаги состояния, такие как флаги равенства, с разветвлением на выходе в качестве условия выполнения команды IOWR(223). в следующем цикле.

В случае, когда Специальный модуль оказывается занятым, операция записи не выполняется. Используйте Флаг равенства для создания удерживающейся программы, как показано ниже, так, чтобы команда IOWR(223) выполнялась в каждом из циклов до выполнения операции чтения.

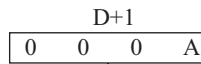
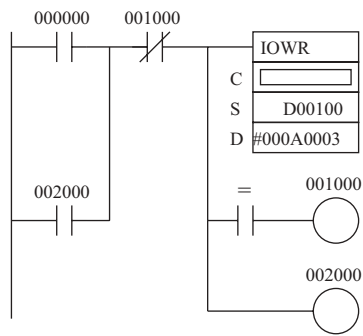


Когда возникают условия ввода, условие поддержания обеспечивается выводом A и командой IOWR(223), выполняемой в каждом из циклов до тех пор, пока флаг равенства не будет переведен в состояние ON. После завершения записи и перевода флага равенства в состояние ON, вывод B переводится в состояние ON и условие поддержания очищается.

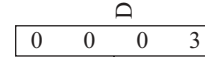
Непосредственно после команд IOWR(223) непременно помещайте флаги состояния, и не вводите эти флаги после любой другой команды. Если флаг состояния располагается после другой команды, он будет подвержен воздействию результата выполнения этой команды.

### Пример

В данном примере команда IOWR(223) используется для записи данных.



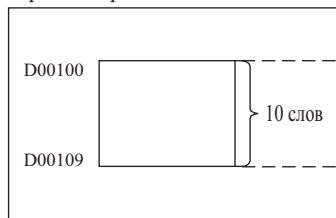
Количество слов,  
подлежащих  
передаче: 10



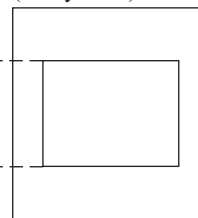
Номер модуля: 3

Код управления (C) изменяется в зависимости от типа  
Специального модуля

Модуль центрального  
процессора



Специальный модуль  
(Модуль #3)



### 3-22 Команды последовательного коммуникационного обмена

В настоящем разделе приводится описание команд, применяемых для выполнения последовательного коммуникационного обмена.

Команда	Мнемоническое изображение	Функциональный код	Страница
PROTOCOL MACRO	PMCR	260	582
TRANSMIT	TXD	236	589
RECEIVE	RXD	235	593
CHANGE SERIAL PORT SETUP	STUP	237	598

#### 3-22-1 Последовательный коммуникационный обмен

Существует два типа команд для осуществления последовательного коммуникационного обмена. Команды TXD(236) – передача, и RXD(235) – прием, осуществляют обмен данными с внешним устройством в режиме обмена данными без протокола. Команда PMCR(260) выполняет прием и передачу данных с использованием протоколов, определяемых пользователем. Различие между командами поясняется в следующей ниже таблице.

**Примечание:** Функция макро протокола PMCR(260) включает команды обмена данными без протокола TXD(236) и RXD(235). При этом команды TXD(236) и RXD(235) используются только для последовательного порта Модуля центрального процессора. Команда PMCR(260) используется только для обслуживания последовательных портов Модуля последовательного коммуникационного обмена и Платы последовательного коммуникационного обмена.

#### TXD(236) и RXD(235)

Могут применяться любые из указанных ниже форматов.

Данные без начального и завершающего кода.



Данные плюс код начала и код окончания.



Данные плюс начальный код.



Данные плюс код окончания CR+LF.



Данные плюс код окончания.



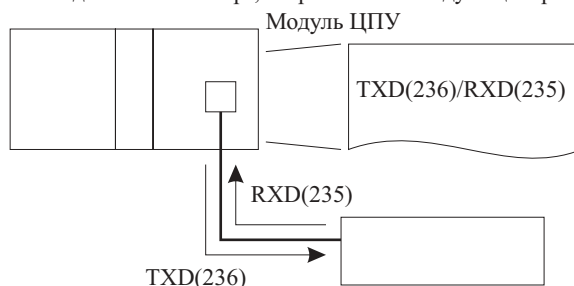
Данные плюс код начала и код окончания CR+LF.



Команды Осуществляют передачу и прием данных только в одном направлении. Можно устанавливать задержку передачи.

#### Режим без протокола (протокол пользователя)

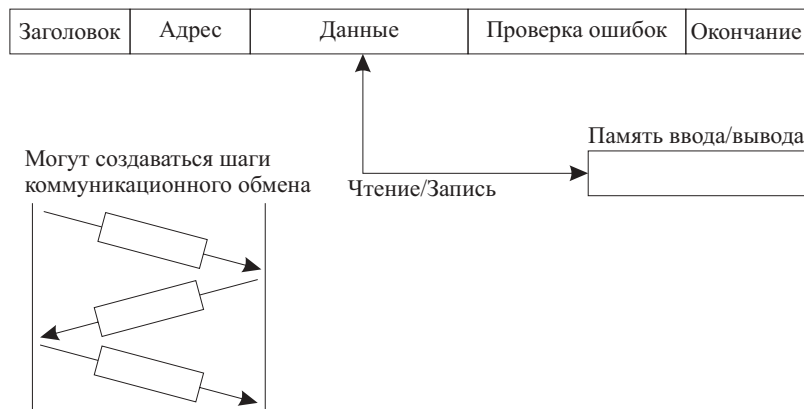
Последовательный порт, встроенный в Модуль центрального процессора



**Примечание:** Команды TXD(236) и RXD(235) не могут использоваться для обслуживания последовательных портов Модулей последовательного коммуникационного обмена или других модулей.

#### PMCR(260)

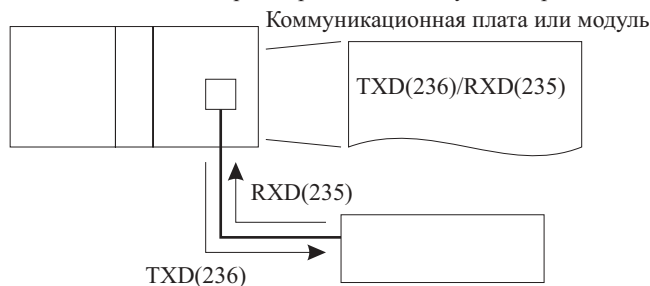
Для достижения соответствия требованиям внешнего устройства могут создаваться следующие типы кадров (сообщений).



- Для передачи и приема данных можно создавать до 16-ти шагов.
- Шаги могут изменяться, а процесс выполнения повторных попыток зависит от принимаемого отклика.
- Можно устанавливать время мониторинга коммуникационного обмена.
- Для программируемого контроллера можно читать и записывать символы.
- Можно использовать повторение символов.
- Другие функции.

#### Режим Макро протокол

Последовательный порт, встроенный в Модуль центрального процессора



### 3-22-2 Команда выполнения макро протокола PROTOKOL MACRO: PMCR(260)

#### Назначение

По команде PMCR(260) производится вызов и выполнение коммуникационной последовательности, зарегистрированной в Плате последовательного коммуникационного обмена или в Модуле последовательного коммуникационного обмена.

#### Символ релейно-контактной схемы

PMCR(260)	
C1	C1: Слово управления 1
C2	C2: Слово управления 2
S	S: Первое передаваемое слово
R	R: Первое принимаемое слово

#### Модификации

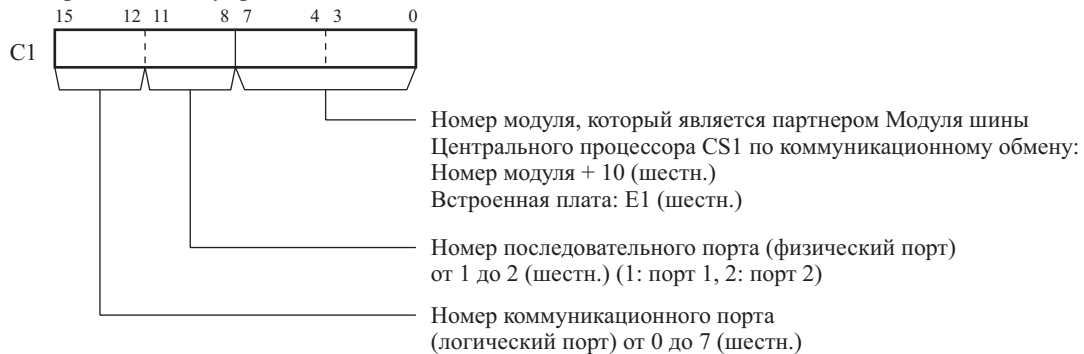
Модификации	Выполняется в каждом цикле для состояния ON	PMCR(260)
	Выполняется один раз для дифференцирования вверх	@ PMCR(260)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****C1: Слово управления 1 и C2: Слово управления 2**

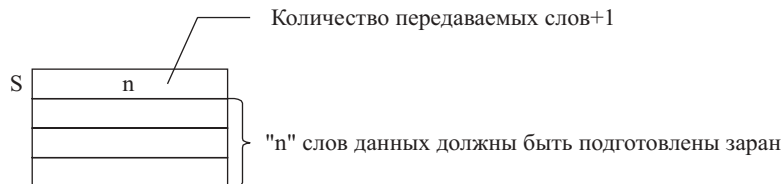
Содержание слов управления показано ниже.

**S: Первое передаваемое слово и область передачи**

Указывается первое слово из последовательности слов, подлежащих передаче. S содержит количество слов, подлежащих передаче + 1 (т.е. включая слово S). Пересылаемые данные начинаются в S+1. Могут пересылаться слова между 0000 и 00FA (от 0 до 250 в десятичном коде).

Если данные для передачи отсутствуют, в качестве константы для S всегда используйте значение 0000.

Если указывается любая другая константа или адрес другого слова, определяется ошибка, флаг ошибки переводится в состояние ON, и команда PMCR(260) не выполняется.

**R: Первое принимаемое слово и область приема**

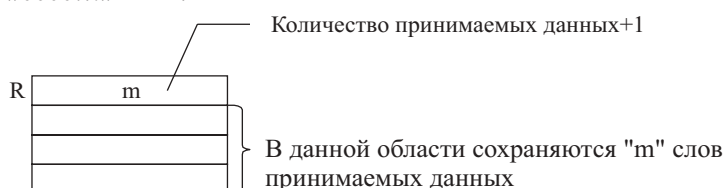
Принимаемые данные автоматически записываются в слова, начиная с R+1. Количество принятых слов плюс 1 (т.е. включая R) автоматически записывается в R между 0000 и 00FA (от 0 до 250 в десятичном коде).

**Необходимые установки перед выполнением команды PMCR**

Установите в m данные (начиная с D), в качестве начальных данных для буфера приема (резервные данные при ошибке приема). Данные m могут устанавливаться в значения 0001 00FA (шестн.) (2 255). В случае, когда в качестве данных m указывается 0000 (шестн.) или 0001 (шестн.), начальное значение приемного буфера сбрасывается в значение, равное 0.

Всегда устанавливайте адрес слова для R, даже если принимаемые данные отсутствуют. Если в данном случае задается константа, определяется ошибка и флаг ошибки переводится в состояние ON, а команда PMCR(260) не выполняется. Если принимаемые данные отсутствуют, R не используется и может применяться для других целей.

В случае, когда принимаемые данные отсутствуют, непременно устанавливайте константу в значение #0000...#FFFF.



### Спецификации операндов

Область	C1	C2	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143			
Рабочая область	W000...W511			
Область удержания бита (Holding bit area)	H000...H511			
Область вспомогательных битов	A000...A447 A448...A959			A000...A959
Область таймера	T0000...T4095			
Область счетчика	C0000...C4095			
Область DM	D00000...D32767			
Область EM, не содержащая банков	E00000...E32767			
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)			
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n = 0...C)			
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)			
Константы	Только указанные значения	0000...03E7 (шестн.) (0...999)	#0000 (двоичное)	
Регистры данных	DR0...DR15	-		
Индексные регистры	-			
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0...-2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)... ,IR15+(++) ,-(-)IR0...,-(-)IR15			

### Описание

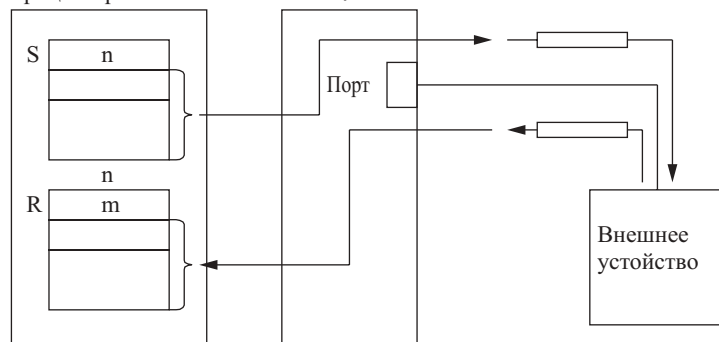
По команде PMCR(260) осуществляется выполнение коммуникационной последовательности, указанной в C2, используя логический порт, указанный в битах 12...15 слова C1, и физический порт, указанный в битах 8...11 слова C1. Адрес модуля указывается в битах 0...7 слова C1.

Если для передачи сообщения в качестве операнда указывается символ, количество передаваемых слов, указанное в S и начиная с S+1, будет принято в качестве области передачи. Если символ указывается в качестве операнда для принимаемого сообщения, принимаемые данные располагаются в памяти, начиная с R+1, а количество принятых слов автоматически записывается в R, если передача завершена без ошибок.

В случае, когда передача успешно не завершена, данные, начиная с R+1 и далее, заданные перед выполнением команды PMCR(260), будут снова прочитаны из приемного буфера и записаны в R+1 и далее.

Модуль центрального процессора

Модуль последовательного коммуникационного обмена



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	<p>Переводится в состояние ON, когда при выполнении команды PMCR(260) для указываемого логического порта флаг включения логического порта находится в состоянии OFF.</p> <p>Переводится в состояние ON, когда данные в слове С выходят за допускаемые пределы. (Флаг ошибки не переводится в состояние ON, если данные слова С2 выходят за допускаемые пределы. Код окончания записывается в код окончания коммуникационного порта (A203...A210) во вспомогательной области).</p> <p>Переводится в состояние ON, когда количество слов в S и R превышает 249 (когда указываются слова).</p> <p>OFF в других случаях.</p>

**Меры предосторожности**

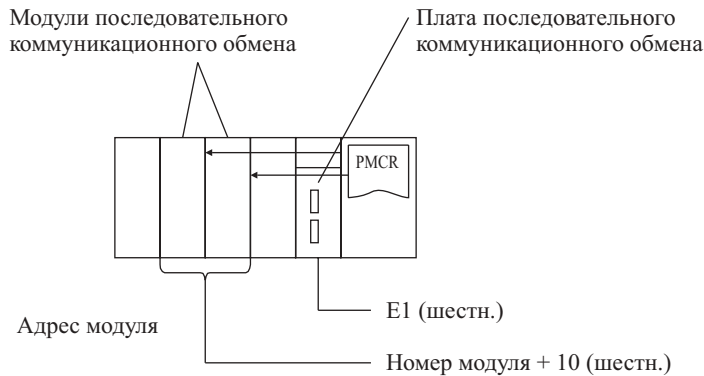
Данные в области передачи, указанные в S, в действительности пересылаются с использованием функции чтения символов R() в пересылаемом сообщении.

В действительности данные принимаются в область принимаемых данных, указанную в R, с использованием функции записи символов W() в принимаемом сообщении.

Для ознакомления с процедурой использования символов R() и W(), обратитесь к Руководству по применению CX-протокола серии CS1.

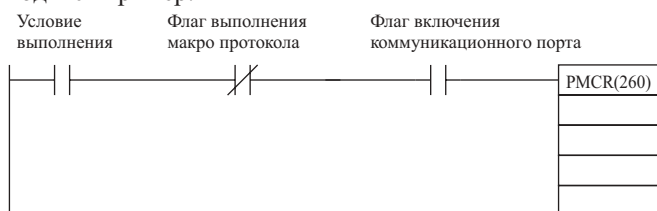
Команда PMCR(260) может выполняться для коммуникационного порта Платы последовательного коммуникационного обмена или Модуля последовательного коммуникационного обмена. В Панель центрального процессора и Панель расширения ввода/вывода может устанавливаться до 16 Модулей последовательного коммуникационного обмена. Адрес модуля/платы, являющегося партнером при проведении обмена данными, должен устанавливаться в битах от 0 до 7 слова С1. Номер последовательного порта должен указываться в битах от 8 до 11. Адреса модулей указываются, как показано в следующей ниже таблице.

Модуль/Плата	Адрес модуля
Плата последовательного коммуникационного обмена	E1 (шестн.)
Модуль последовательного коммуникационного обмена	Номер модуля + 10 (шестн.)



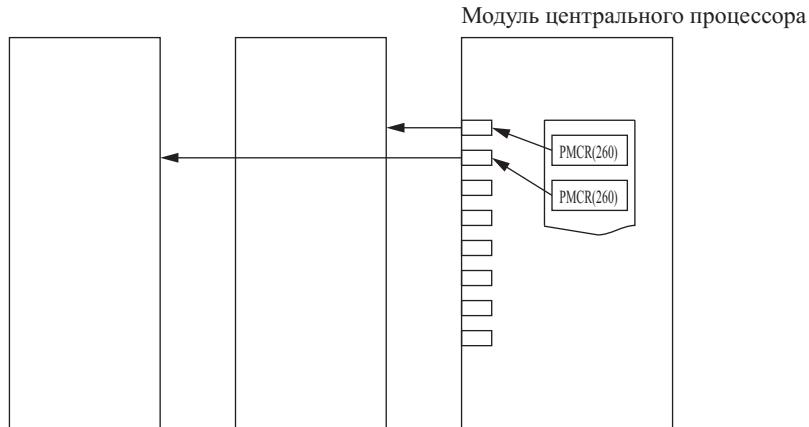
Соответствующий флаг выполнения макро протокола переводится в состояние ON в начале выполнения команды PMCR(260). После завершения выполнения коммуникационной последовательности и записи данных в указанную область принимаемых данных этот флаг переводится в состояние OFF.

A.N.C ввод для соответствующего флага выполнения макро протокола должен использоваться в качестве части условия выполнения при выполнении команды PMCR(260) для того, чтобы для одного коммуникационного порта за один раз выполнялась только одна коммуникационная последовательность. Ниже приводится пример.





Команды SEND(090), RECV(098) и CMND(490) также используют логические порты 0...7 для передачи коммуникационных последовательностей через Платы и Модули последовательного коммуникационного обмена (используя FINS команды). Команда PMCR(260) не может выполняться для логического порта, который уже используется командами SEND(090), RECV(098), CMND(490) или PMCR(260). Для предотвращения одновременной передачи коммуникационной последовательности через один и тот же порт, соответствующий флаг включения коммуникационного порта (A20200 A20207) должен использоваться в качестве N.O. ввода в условии выполнения команды PMCR(260), как показано на рисунке выше.



Флаг ошибки переводится в состояние ON в следующих случаях.

- Когда при выполнении команды PMCR(260) соответствующий флаг включения коммуникационного порта находится в состоянии OFF.
- Когда С1 находится за пределами допустимого диапазона значений.

#### Указание области приема

Перед выполнением команды PMCR(260) пользователь обязан задать резервные данные в области приема данных, для того, чтобы обеспечивать прием данных при возникновении ошибок. Когда команда PMCR(260) выполняется, данные в буфере приема автоматически заносятся в область приема. Одним из применений резервных данных является следующее: некоторое значение (резервные данные) предварительно задаются таким образом, что текущее значение не будет читаться как нулевое значение при возникновении ошибки передачи в процессе выполнения протокола и чтения текущего значения контроллера.

#### Относящиеся к команде флаги и слова

При выполнении команды PMCR(260) могут использоваться следующие ниже флаги и слова.

#### Вспомогательная область

Наименование	Адрес	Содержание
Флаг включения коммуникационного порта	A20200...A20207	Переводится в состояние ON, когда выполнение сетевого коммуникационного обмена разрешается (включая команду PMCR(260)). Биты от 00 до 07 соответствуют логическим портам от 0 до 7 соответственно. Флаг включения сетевого коммуникационного обмена переводится в состояние OFF, когда коммуникационный обмен начинает выполняться и переводится в состояние ON, когда обмен завершается (вне зависимости от наличия ошибок или их отсутствия при выполнении обмена).
Флаг ошибки коммуникационного порта	A21900...A21907	Переводится в состояние ON при возникновении ошибки выполнения сетевого обмена данными. Биты от 00 до 07 соответствуют логическим портам от 0 до 7 соответственно. Состояние флага поддерживается до начала следующего сетевого обмена данными. Флаг переводится в состояние OFF, когда начинается следующий обмен данными, даже если при последнем выполнении произошла ошибка.

Наименование	Адрес	Содержание
Коды завершения коммуникационного порта	A203...A210	Содержит код завершения, записываемый после выполнения сетевого обмена данными. Слова A203...A210 соответствуют логическим портам от 0 до 7 соответственно. Код завершения равен 00 в процессе выполнения команды коммуникационного обмена. После завершения обмена записывается новый код отклика. Содержание этих слов при запуске сбрасывается.

#### Отклики при коммуникационном обмене

Код	Содержание
1106 (шестн.)	Отсутствует номер соответствующей программы. Указан номер последовательности передачи/приема, который не зарегистрирован. Измените номер последовательности передачи/приема или добавьте номер, используя CX-программатор.
2201 (шестн.)	Не может выполняться вследствие выполнения протокола. Вследствие того, что макро протокол уже был выполнен, дальнейшее его выполнение не принимается. Введите дополнительно в программу условие NC (сигнала подключения к сети) для флага выполнения макро протокола.
2202 (шестн.)	Не может выполняться вследствие остановки. Вследствие того, что протокол включен, дальнейшее выполнение не принимается. Введите дополнительно в программу условие NC (сигнала подключения к сети) для флага изменения установок последовательного обмена.
2401 (шестн.)	Отсутствует таблица регистрации. В данных макро протокола или в переданных данных присутствует ошибка. Осуществите передачу данных макро протокола, используя CX- программатор.
Прочие	Для ознакомления с прочими кодами отклика обратитесь к Руководству по применению команд коммуникационного обмена серии CS1 (W342).

#### Область встроенной платы

Наименование	Адрес	Содержание
Флаг выполнения макро протокола, порт 1.	СЮ 190915	Переводится в состояние ON при выполнении команды PMCR(260). Флаг оставляется в состоянии OFF, если выполнение команды не заканчивается успешно. Флаг переводится в состояние OFF, когда выполнение команды завершается (либо заканчивается либо отменяется).
Флаг выполнения макро протокола, порт 2.	СЮ 191915	

#### Область Модуля шины центрального процессора CS1

$N = 1500 + 25 \times \text{номер модуля.}$

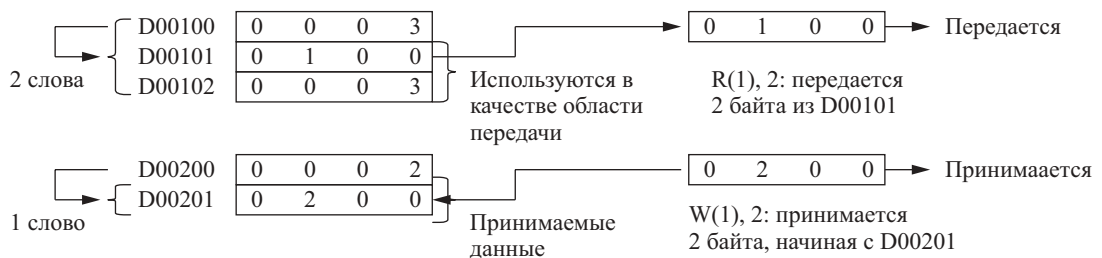
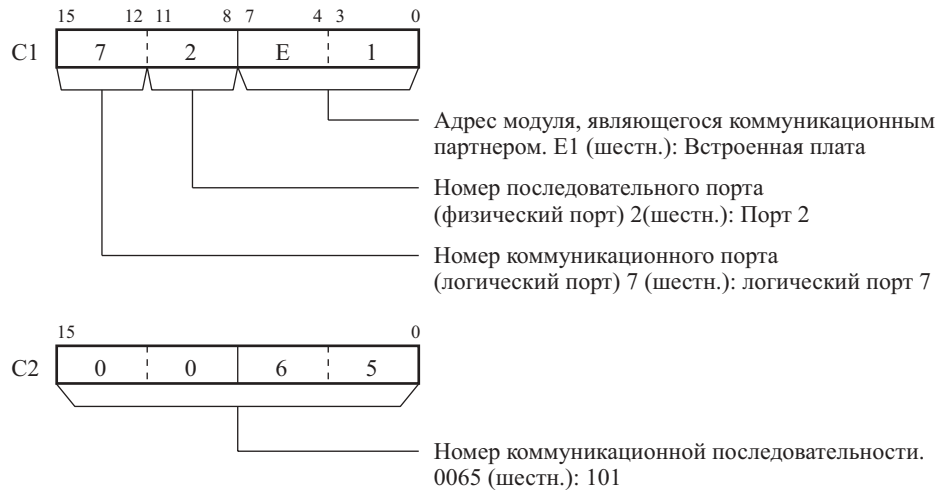
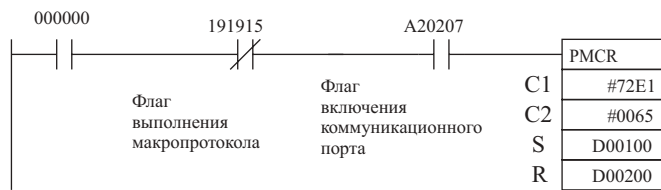
Наименование	Адрес	Содержание
Флаг выполнения макро протокола, порт 1.	Бит 15 слова СЮ n + 9	Переводится в состояние ON при выполнении команды PMCR(260). Флаг оставляется в состоянии OFF, если выполнение команды не заканчивается успешно.
Флаг выполнения макро протокола, порт 2.	Бит 15 слова СЮ n + 19	Флаг переводится в состояние OFF, когда выполнение команды завершается (либо заканчивается либо отменяется).

#### Примеры

В следующем ниже примере, когда СЮ 0000 находится в состоянии ON, последовательность коммуникационного обмена № 101 (0065 шестн.) выполняется до тех пор, пока флаг включения коммуникационного порта 7 (A20207) находится в состоянии ON, а флаг выполнения макро протокола через порт 1 (СЮ 190915) находится в состоянии OFF.

Если в передаваемом сообщении указывается операнд для символа, 2 слова данных, начиная с D00101, используются в качестве области передачи (так как содержанием D00100 является #0003).

Если в принимаемом сообщении указывается операнд для символа, производится запись 2-х слов данных, начиная с D00201, а количество принятых слов +1 записывается в D00200.



**Примечание:** Как показано выше символьная функция чтения  $R()$ , в передаваемом сообщении или символьная функция записи  $W()$ , в действительности передает или принимает данные.

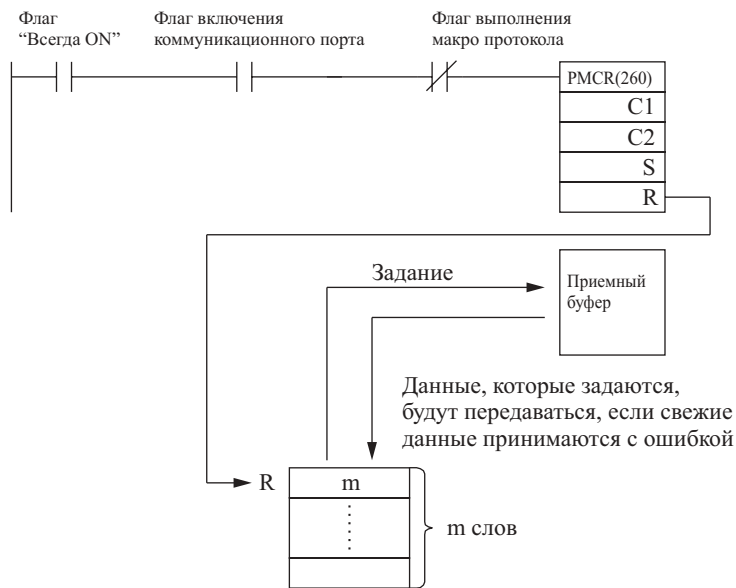
### Удержание области приема

Содержание приемного буфера сбрасывается в нули непосредственно перед выполнением коммуникационной последовательности по команде  $PMCR(260)$ . Если для периодического чтения текущего значения и других данных применяется программа, подобная приведенной ниже программе, и данные не могут читаться вследствие ошибки приема или вследствие других причин, прочитанные данные очищаются до следующего безошибочного чтения.

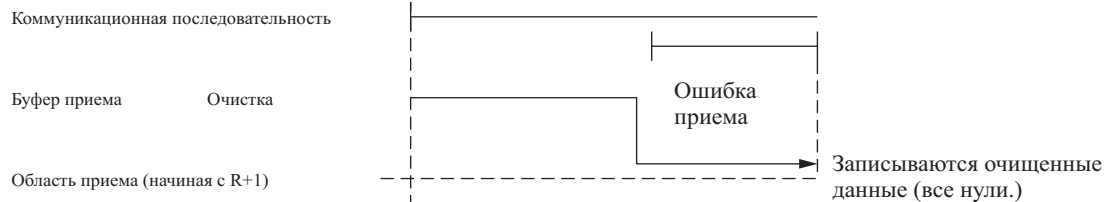
Для удержания данных в области приема в том случае, когда происходит ошибка приема, существует специальная функция. При использовании такой функции данные из первых  $m$  слов области приема передаются в буфер приема после его очистки, однако, перед выполнением коммуникационной последовательности. Это предотвращает область приема от временного сброса посредством записи последних принятых данных, когда новые данные не могут успешно приниматься.

Укажите количество слов области приема, которое должно поддерживаться, в виде значения  $m$ . Если задается значение 0 или 1, функция удержания отключается, и данные области приема сбрасываются в нулевое значение.

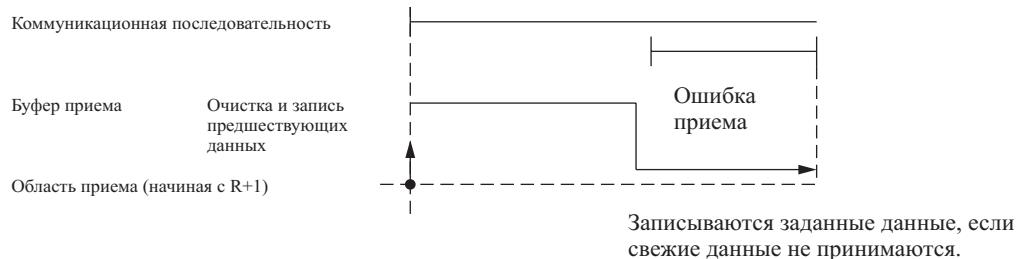
Следующий ниже пример программы показывает команды, используемые для постоянного или периодического выполнения команды  $PMCR(260)$  для чтения данных с использованием простоя операции приема.



**Область приема не удерживается**



**Область приема удерживается**



3-22-3 Команда выполнения передачи данных Transmit: TXD(236)

**Назначение**

По команде TXD(236) производится вывод указанного количества байтов данных в порт RS-232C, Модуля центрального процессора.

**Символ релейно-контактной схемы**

— TXD(236)	
S	S: Первое исходное слово
C	C: Слово управления
N	N: Количество байтов от 0000 до 0100 (шестн.)(от 0 до 256)

**Модификации**

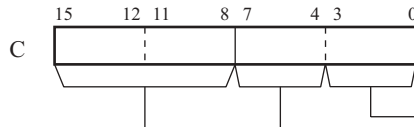
Модификации	Выполняется в каждом цикле для состояния ON	TXD(236)
	Выполняется один раз для дифференцирования вверх	@ TXD(236)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды**

Содержание слова управления С показано ниже.



Порядок байтов.

0: Вначале наиболее значимые байты.

1: Вначале наименее значимые байты

Сигналы управления RS и ER.

0: Управление сигналами RS и ER не выполняется.

1: Управление сигналом RS.

2: Управление сигналом ER.

3: Управление сигналами RS и ER

не используются

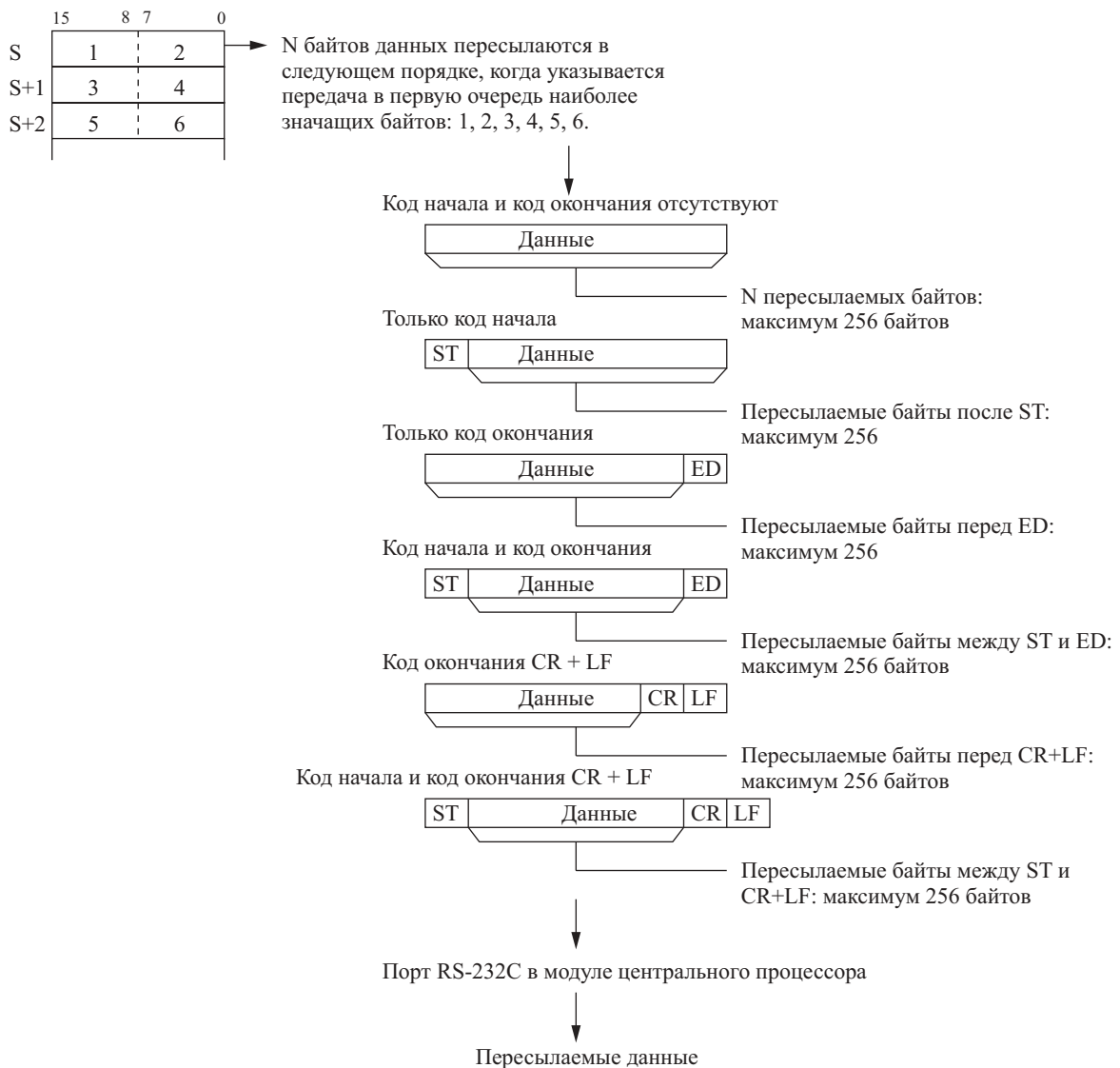
**Спецификации операндов**

Область	C	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A447 A448...A959		
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	—	Только указанные значения	#0000...#0100 (двоичные)
Регистры данных	—	DR0...DR15	
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048 +2047,IR0...-2048...+2047,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-(-)IR15		

**Описание**

По команде TXD(236) производится чтение N байтов данных из слов S...S+N2) –1 и вывод данных через порт RS-232C Модуля центрального процессора в режиме передачи без протокола и без преобразования данных. При выводе данных добавляются код начала и код окончания, указанные в начальных установках для режима обмена без протокола.

Передача данных возможна только когда флаг готовности к передаче (A39205) находится в состоянии ON. На следующем ниже рисунке показан порядок, в котором данные передаются и содержание кадра для различных установок.



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	<p>Переводится в состояние ON, когда режим обмена без протокола не задан в начальных установках Программируемого контроллера.</p> <p>Переводится в состояние ON, когда значение С находится за пределами допустимых значений.</p> <p>Переводится в состояние ON, когда значение N находится за пределами значений от 0000 до 0100 (шестн.)</p> <p>Переводится в состояние ON, когда попытка передачи производится в момент, когда флаг готовности к передаче находится в состоянии OFF.</p> <p>Переводится в состояние OFF в других случаях.</p>

### Меры предосторожности

Команда TXD(236) может применяться только для порта RS-232C Модуля центрального процессора и только тогда, когда этот порт установлен в режим обмена без протокола.

Для передачи сообщений в начальных установках может быть задан формат кадра.

- Код начала: Отсутствует или от 00 до FF (шестн.).
- Код окончания: Отсутствует, CR + LF или от 00 до FF (шестн.).

Данные будут пересылаться с любыми кодами начала и/или окончания, указанными в начальных установках Программируемого контроллера. При указании кода начала и кода окончания, максимальное количество байтов, задаваемых в качестве значения для N, будет уменьшено до 254 байтов.

Данные могут передаваться только тогда, когда флаг готовности к передаче (A38205) находится в состоянии ON.

Данные передаются в соответствии с порядком, указанным в С.

В случае, когда в качестве значения для N указывается 0, передача данных не производится.

Если в слове С указывается использование контрольного сигнала RS, бит 15 слова S будет использоваться в качестве RS сигнала. Если в слове С указывается использование контрольного сигнала ER, бит 15 слова S будет использоваться в качестве ER сигнала. Если в слове С указывается использование контрольных сигналов RS и ER, бит 15 слова S будет использоваться в качестве RS сигнала, а бит 14 слова S будет использоваться в качестве сигнала ER.

Если в слове С для контрольных сигналов RS и ER заданы значения 1, 2, или 3, команда TXD(236) будет выполняться вне зависимости от состояния сигнала готовности к передаче (A39205).

В следующих ниже случаях определяется ошибка, и флаг ошибки переводится в состояние ON.

- В начальных установках Программируемого контроллера порту не задан режим обмена без протокола.
- Значение С находится за пределами допускаемых значений.
- Значение N выходит за пределы значений от 0000 до 0100 (шестн.).
- Предпринимается попытка передачи данных в момент, когда флаг готовности к передаче (A39205) находится в состоянии OFF.

Некоторые устройства, в адрес которых пересылаются данные, могут требовать некоторой задержки передачи. В этом случае задавайте требуемую внешним устройством величину задержки.

#### Флаги и слова, относящиеся к команде

Следующие ниже параметры начальных установок Программируемого контроллера и флаги вспомогательной области могут использоваться при выполнении команды TXD(236).

Начальные установки Программируемого контроллера

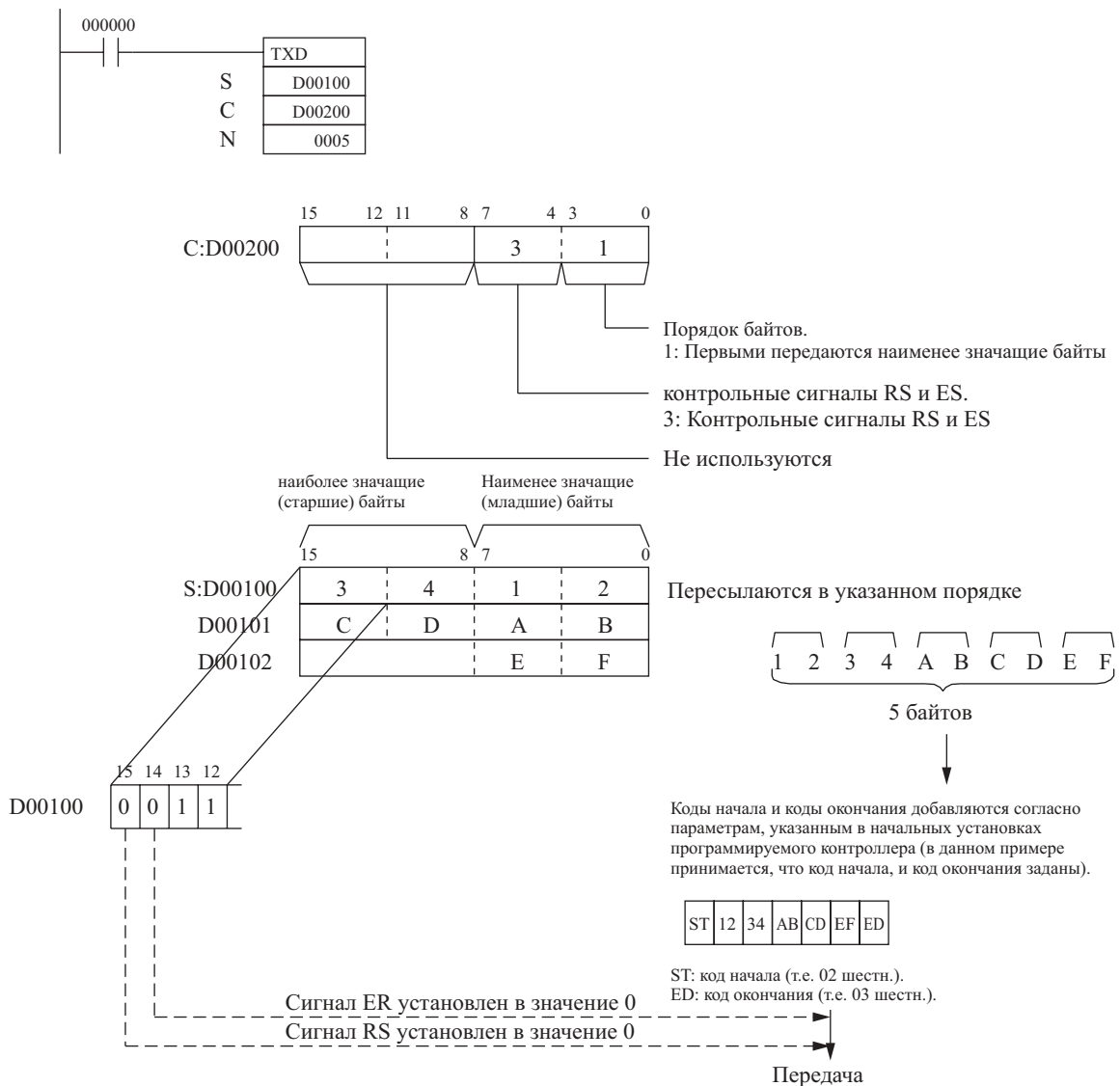
Наименование	Содержание	Установки
Код начала для режима обмена без протокола	Указывается, если код начала должен использоваться в формате кадра для коммуникационного обмена без протокола.	0: Не указывается (по умолчанию). 1: Да (задается от 00 до FF (шестн.)).
Код окончания для режима обмена без протокола	Указывается, если код окончания должен использоваться в формате кадра для коммуникационного обмена без протокола.	0: Не указывается (по умолчанию). 1: Да (задается от 00 до FF (шестн.) или CR+LF).
Задержка передачи в режиме обмена без протокола	Указанные данные будут передаваться из порта только по истечению заданной задержки от момента выполнения команды TXD(260) в релейно-контактной схеме.	От 0 до 99990 мсек. (в единицах, равных 10 мсек.). По умолчанию: 0 мсек.

#### Вспомогательная область

Наименование	Адрес	Содержание
Флаг готовности порта RS-232C к передаче.	A39205	Переводится в состояние ON, когда данные могут пересылаться в режиме обмена без протокола.

#### Примеры

В следующем ниже примере, когда СЮ 000000 находится в состоянии ON, 5 байтов данных, начиная с D00100, пересылаются без преобразования, как показано ниже.



### 3-22-4 Команда выполнения приема данных RECEIVE: RXD(235)

#### Назначение

По команде RXD(235) производится чтение указанного количества байтов данных в указанном порте.

#### Символ релейно-контактной схемы

RXD(235)	
D	D: Первое слово назначения
C	C: Слово управления
N	N: Количество байтов от 0000 до 0100 (шестн.) (От 0 до 256 в десятичном коде)

#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	RXD(235)
	Выполняется один раз для дифференцирования вверх	@ RXD(235)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

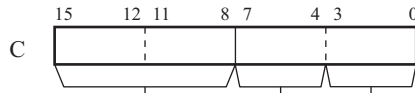


**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды**

Содержание слова управления С показано ниже.



Порядок байтов.

0: Вначале читаются наиболее значимые байты.

1: Вначале читаются наименее значимые байты

Мониторинг сигналов CS и DR.

0: Мониторинг сигналов CS и DR не выполняется.

1: Мониторинг сигнала CS .

2: Мониторинг сигнала DR.

3: Мониторинг сигналов CS и DR

Не используется

**Спецификации операндов**

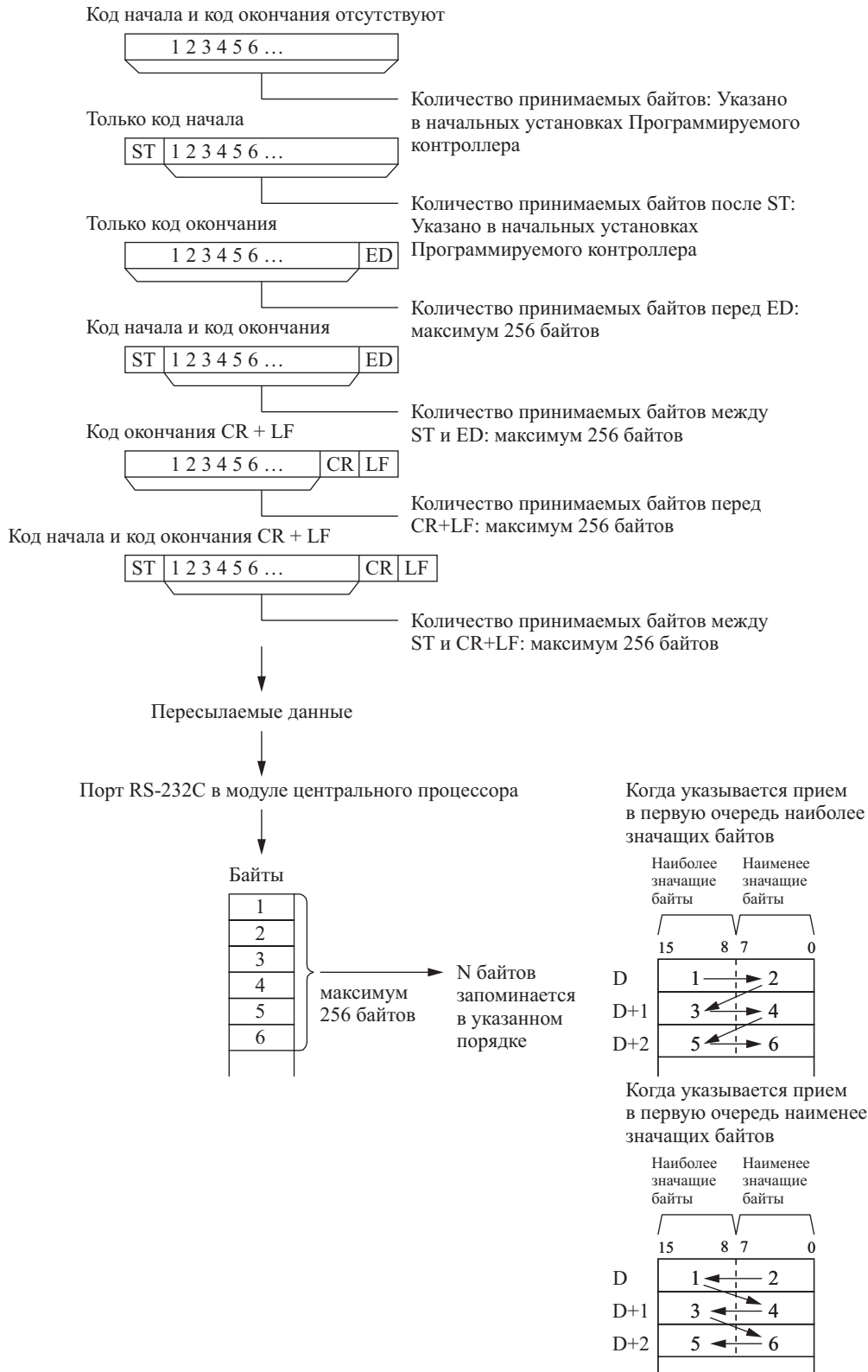
Область	D	C	N
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A448...A959	A000...A447 A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	–	Только указанные значения	#0000...#0100 (двоичные)
Регистры данных	–	DR0...DR15	
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048 +2047,IR0...-2048...+2047,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15(++)) ,-(-)IR0...,(-)IR15		

**Описание**

По команде RXD(235) производится чтение данных, которые принимаются в режиме обмена без протокола из порта RS-232C в Модуле центрального процессора, и запись N байтов данных в слова от D до D + (N 2) – 1. При приеме данных также принимаются код начала и код окончания, указанные в начальных установках Программируемого контроллера для режима обмена без протокола. В случае, когда все N байтов данных не принимаются, производится запись байтов, которые были приняты.

Прием данных возможен только тогда, когда флаг готовности к приему (A39206) находится в состоянии ON.

На следующем ниже рисунке показан порядок, в котором данные принимаются и содержание принимаемого кадра для различных установок.



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда режим обмена без протокола не задан в начальных установках Программируемого контроллера. Переводится в состояние ON, когда значение С находится за пределами допустимых значений. Переводится в состояние ON, когда значение N находится за пределами значений от 0000 до 0100 (шестн.) Переводится в состояние OFF в других случаях.

**Меры предосторожности**

Команда RXD(235) может применяться только для порта RS-232C Модуля центрального процессора и только тогда, когда этот порт установлен в режим обмена без протокола.

Для приема сообщений в начальных установках может быть задан формат кадра.

- Код начала: Отсутствует или от 00 до FF (шестн.).
- Код окончания: Отсутствует, CR + LF или от 00 до FF (шестн.). Если код окончания не указан, количество байтов, подлежащих приему, устанавливается от 00 до FF (шестн.) (от 1 до 256 в десятичном коде; 00 используется для задания 256 байтов.)

Флаг завершения приема (A39206) переводится в состояние ON после завершения приема количества байтов, указанного в начальных установках Программируемого контроллера. Когда флаг завершения приема переводится в состояние ON, количество байтов в счетчике приема (A393) будет иметь значение равное заданному значению, указанному в начальных установках Программируемого контроллера. В случае приема данных, превышающих заданное значение, флаг переполнения при приеме (A39207) будет переведен в состояние ON.

В случае, когда код окончания задан в начальных установках Программируемого контроллера, флаг завершения приема (A39206) переводится в состояние ON после принятия кода окончания или после приема 256 байтов. Если после перевода флага завершения приема (A39206) принимается дополнительное количество байтов, флаг переполнения при приеме (A39207) будет переведен в состояние ON.

После выполнения команды RXD(235) данные записываются в памяти, начиная с D, флаг завершения приема (A39206) переводится в состояние OFF (даже если флаг переполнения при приеме (A39207) переводится в состояние ON), и счетчик приема (A393) обнуляется.

Если бит перезапуска порта RS-232C (A52600) переводится в состояние ON, флаг завершения приема (A39206) переводится в состояние OFF (даже если флаг переполнения (A39207) находится в состоянии ON), и счетчик приема (A393) обнуляется.

Данные принимаются в соответствии с порядком, указанным в С.

Если в начальных установках Программируемого контроллера заданы код начала и код окончания, количество байтов, указанных в качестве значения для N, будет игнорироваться, и при выполнении команды RXD(235) данные, начиная с кода начала, и до кода окончания, будут записаны в слова, начиная с D.

В случае, когда в качестве значения для N указывается 0, флаг завершения приема (A39206) будет переведен в состояние OFF, счетчик приема (A393) будет обнулен, и запись данных в память не производится.

Если в слове С задан мониторинг сигнала CS, состояние сигнала CS будет записано в бит 15 слова D.

Если в слове С задан мониторинг сигнала DR, состояние сигнала DR будет записано в бит 15 слова D.

Если в слове С задан мониторинг сигналов CS и DR, состояние сигнала CS будет записано в бит 15 слова D, а состояние сигнала DR будет записано в бит 14 слова D.

Принимаемые данные не будут записаны, если задан мониторинг сигнала CS или DR.

Если в слове С для контрольных сигналов RS и ER заданы значения 1, 2, или 3, команда RXD(235) будет выполняться вне зависимости от состояния сигнала готовности к приему (A39206).

В следующих ниже случаях определяется ошибка, и флаг ошибки переводится в состояние ON.

- В начальных установках Программируемого контроллера порту не задан режим обмена без протокола.
- Значение С находится за пределами допускаемых значений.
- Значение N выходит за пределы значений от 0000 до 0100 (шестн.).

До выполнения чтения данных, принимаемых по команде RXD(235) другие данные приниматься не могут. Выполняйте чтение данных с момента перевода флага завершения приема в состояние ON и до следующего выполнения команды RXD(235).

**Флаги и слова, относящиеся к команде**

Следующие ниже параметры начальных установок Программируемого контроллера и флаги вспомогательной области могут использоваться при выполнении команды RXD(235).

**Начальные установки Программируемого контроллера**

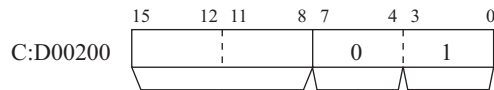
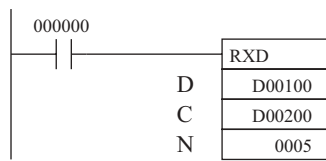
Наименование	Содержание	Установки
Код начала для режима обмена без протокола.	Указывается, если код начала должен использоваться в формате кадра для коммуникационного обмена без протокола.	0: Не указывается (по умолчанию). 1: Да (задается от 00 до FF (шестн.)).
Код окончания для режима обмена без протокола.	Указывается, если код окончания должен использоваться в формате кадра для коммуникационного обмена без протокола.	0: Не указывается (по умолчанию). 1: Да (задается от 00 до FF (шестн.) или CR+LF).
Количество принимаемых байтов в режиме обмена без протокола.	Указывает количество байтов, подлежащих чтению, когда код окончания не указан.	От 01 до FFFF (шестн.) (от 1 до 255 в десятичном коде). По умолчанию: 00 (256 байтов).

**Вспомогательная область**

Наименование	Адрес	Содержание
Флаг завершения приема порта RS-232C.	A39206	Переводится в состояние ON, когда прием в режиме обмена данных без протокола завершен. Количество принимаемых байтов указано: Флаг переводится в состояние ON, когда завершен прием указанного количества байтов. Код окончания указан: Флаг переводится в состояние ON, когда принимается код окончания или завершен прием 256 байтов данных.
Флаг переполнения порта RS-232C при приеме.	A39207	Переводится в состояние ON, когда количество принятых данных превышает указанное количество данных. Количество принимаемых данных указано: Флаг переводится в состояние ON, когда после завершения приема указанного количества данных перед выполнением следующей команды RXD(235) принимаются дополнительные данные. Код окончания указан: Флаг переводится в состояние ON, когда после приема кода окончания перед выполнением следующей команды RXD(256) принимаются дополнительные данные или осуществляется прием 257-го байта до приема кода окончания.
Счетчик приема по порту RS-232C	A393	Осуществляет подсчет количество принимаемых байтов в режиме обмена без протокола (в шестнадцатеричном коде).

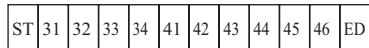
**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, 5 байтов данных принимаются через порт RS-232C, и запоминаются в словах, начиная с D00100.



Порядок байтов.  
 1: Первыми передаются наименее значащие байты  
 Контрольные сигналы RS и ES.  
 Мониторинг контрольных сигналов CS и DR.  
 0: Мониторинг сигналов CS и DR не производится  
 Не используются

В данном примере принимается, что в начальных установках Программируемого контроллера указаны код начала и код окончания.



ST: код начала (т.е. 02 шестн.).  
 ED: код окончания (т.е. 03 шестн.).

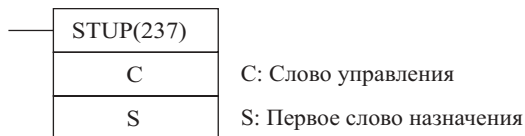
	наиболее значащие (старшие) байты				Наименее значащие (младшие) байты			
	15	8	7	0	15	8	7	0
S:D00100	3	2	3	1				
D00101	3	4	3	3				
D00102	4	2	4	1				
D00103	4	4	4	3				
D00104	4	6	4	5				

### 3-22-5 Команда изменения установок последовательного порта CHANGE SERIAL PORT SETUP: STUP(237)

#### Назначение

По команде STUP(237) производится изменение коммуникационных параметров последовательного порта Модуля центрального процессора (Модуля шины центрального процессора CS1), или Платы последовательного коммуникационного обмена.

#### Символ релейно-контактной схемы



#### Модификации

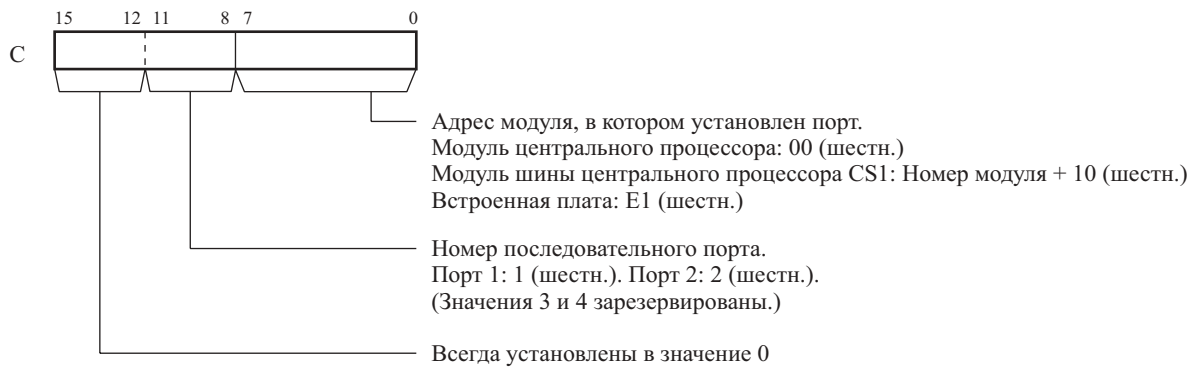
Модификации	Выполняется в каждом цикле для состояния ON	STUP(237)
	Выполняется один раз для дифференцирования вверх	@ STUP(237)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Не применяются

#### Операнды

Содержание слова управления C показано ниже.



### Спецификации операндов

Область	C	S
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6134
Рабочая область	W000...W511	W000...W502
Область удержания бита (Holding bit area)	H000...H511	H000...H502
Область вспомогательных битов	A000...A438 A448...A959	A000...A438 A448...A950
Область таймера	T0000...T4095	T0000...T4086
Область счетчика	C0000...C4095	C0000...C4086
Область DM	D00000...D32767	D00000...D32758
Область EM, не содержащая банков	E00000...E32767	E00000...E32758
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32758 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	Только указанные значения	#0000
Регистры данных	DR0...DR15	
Индексные регистры	-	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0...-2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-)IR15	

### Описание

По команде STUP(237) производится запись 10 слов данных в слова S...S+9 области установок коммуникационного обмена Модуля, имеющего указанный адрес, как показано в следующей ниже таблице. Когда слову S присваивается значение константы #0000, установки коммуникационного обмена соответствующего порта устанавливаются в значение по умолчанию.

639

Адрес модуля	Модуль	Номер последовательного порта	Последовательный порт	Область установок коммуникационного обмена последовательного порта
00 (шестн.)	Модуль центрального процессора	1 (шестн.)	Порт 1	Коммуникационные параметры периферийного порта в начальных установках Программируемого контроллера.
		2 (шестн.)	Порт 2	Коммуникационные параметры порта RS-232C в начальных установках Программируемого контроллера.

Адрес модуля	Модуль	Номер последовательного порта	Последовательный порт	Область установок коммуникационного обмена последовательного порта
Номер модуля +10 (шестн.)	Модуль последовательного коммуникационного обмена (Модуль шины центрального процессора CS1)	1 (шестн.)	Порт 1	10 слов, начиная с D30000 +100...номер Модуля.
		2 (шестн.)	Порт 2	10 слов, начиная с D30000 +100...номер Модуля +10.
E1 (шестн.)	Плата последовательного коммуникационного обмена (Встроенная плата)	1 (шестн.)	Порт 1	10 слов, начиная с D32000.
		2 (шестн.)	Порт 2	10 слов, начиная с D32010

При выполнении команды STUP(237) флаг изменения параметров соответствующего порта (A61901, A61902, A619...A636) переводится в состояние ON. Флаг остается в этом состоянии до завершения выполнения изменения параметров.

Используйте команду STUP(237) для изменения коммуникационных параметров порта для выполнения действий, базирующихся на особых условиях. Например, команда STUP(237) может использоваться для перехода к режиму обмена Host Link для осуществления мониторинга и программирования из главного компьютера при возникновении заданных условий в процессе выполнения коммуникационных последовательностей действий с подключенным модемом.

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение С находится за пределами допустимых значений. Переводится в состояние ON, когда команда STUP(237) выполняется для порта, у которого флаг изменения коммуникационных параметров находится в состоянии ON. Переводится в состояние ON, когда команда STUP(237) выполняется в задаче прерывания. Переводится в состояние OFF в других случаях.

#### Меры предосторожности

Коммуникационные параметры состоят из протокола обмена, скорости обмена, формата данных (метода передачи макро протокола и максимальной длины коммуникационных данных макро протокола) и других параметров. Для детального ознакомления обратитесь к *Руководству по эксплуатации Программируемых контроллеров серии CS1 (W339)* или *Руководству по эксплуатации Платы последовательного коммуникационного обмена и Модуля последовательного коммуникационного обмена (W336)*.

#### Флаги и слова, относящиеся к команде

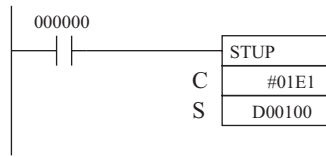
Следующие ниже флаги могут использоваться при выполнении команды STUP(237). Эти флаги расположены во вспомогательной области.

Наименование	Адрес	Содержание
Флаг изменения коммуникационных параметров периферийного порта.	A61901	Переводится в состояние ON при изменении коммуникационных параметров периферийного порта.
Флаг изменения параметров порта RS-232C.	A61902	Переводится в состояние ON при изменении коммуникационных параметров порта RS-232C.
Флаги изменения параметров портов 1...4 Модулей последовательного коммуникационного обмена 1...15	От A620 бит 01...04 до A635 бит 01...04	Переводится в состояние ON при изменении коммуникационных параметров порта Модуля последовательного коммуникационного обмена.
Флаги изменения параметров портов 1...4 Плат последовательного коммуникационного обмена 1...15	От A63601 до A63604	Переводится в состояние ON при изменении коммуникационных параметров порта Платы последовательного коммуникационного обмена.

#### Примеры

В следующем ниже примере, когда CIO 000000 переводится в состояние ON, коммуникационные параметры последовательного порта 1 Платы последовательного коммуникационного обмена (Встроенной

платы) изменяются в значения, содержащиеся в 10-ти словах от D00100 до D00109. В данном примере установки изменяют режим протокола обмена на режим макро протокола.



S: D00100	0	6	0	0
S+1: D00101	0	0	0	0
S+2: D00102				
S+9: D00109				

Установки порта: По умолчанию,  
Режим протокола: 6 (шестн.) (макро протокол).  
Скорость обмена: По умолчанию (9600 б/сек).

↓ Преобразование

Слова области DM, распределенные начальным установкам коммуникационного обмена Платы последовательного коммуникационного обмена.

D32000	0	6	0	0
D32001	0	0	0	0
D32002				
D32009				



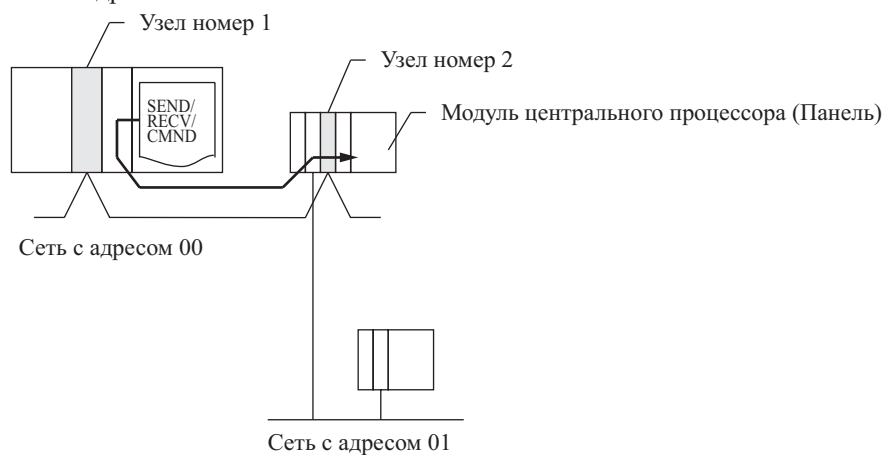
## 3-23 Сетевые команды

Сетевые команды можно подразделить на два типа: SEND(090)/RECV(098) и CMND(490). Эти команды передаются в сети между модулями (Модулями центрального процессора, Модулями шины центрального процессора CS1 и компьютерами) для передачи данных и управления работой, например, для изменения режима работы.

Команда	Содержание сообщения
SEND(090)/ RECV(098)	Команды передачи /приема данных (FINS команды).
CMND(490)	Произвольно выбранные команды (FINS команды).

Команды, выполняемые при подаче сетевых команд, известны как «FINS» команды. Эти команды используются для коммуникационного обмена данными между устройствами управления промышленной автоматикой. С помощью FINS команд можно осуществлять коммуникационный обмен данными (в формате команда/ отклик) с любым Модулем в любой сети или собственно в Панели центрального процессора посредством указания адреса сети, адреса узла, или адреса желаемого модуля.

В следующем ниже примере, команда FINS посылается в Модуль центрального процессора через узел 2 в сети с адресом 00.



1,2,3...

## 1. Адрес сети:

Адрес сети (локальная сеть = 00).

## 2. Номер узла.

Логический адрес в сети.

## 3. Номер модуля.

Номер Модуля назначения.

• Модуль центрального процессора: 00

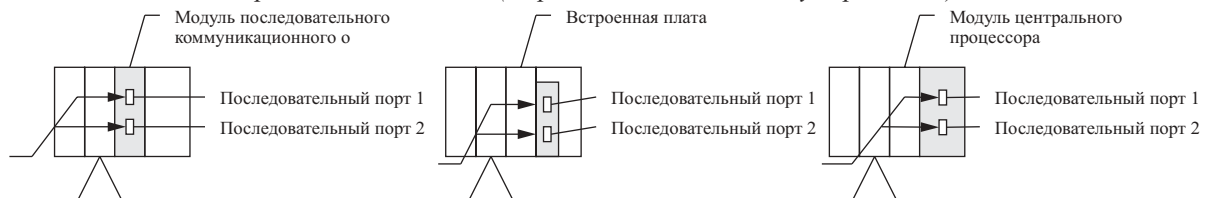
• Модуль шины центрального процессора: Номер модуля + 10 (шестн.).

• Специальный модуль CS1: Номер модуля + 20 (шестн.).

• Встроенная плата: E1 (шестн.).

• Компьютер: 01.

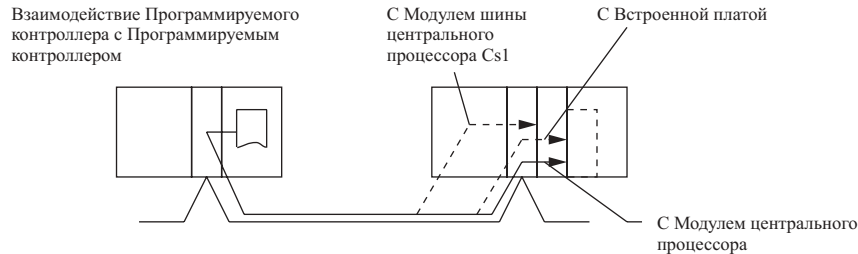
**Примечание:** Можно также указывать последовательный порт (от 1 до 4), находящийся в устройстве назначения. (Порт 0 обозначает само устройство.)

**Образцы сетевого коммуникационного обмена**

Следующие ниже примеры показывают три типа сетевого коммуникационного обмена: взаимодействие Программируемого контроллера с другими устройствами в сети, взаимодействие Программируемого контроллера с последовательными портами других устройств в сети, и коммуникационный обмен с главным компьютером, подключенным в Host Link.

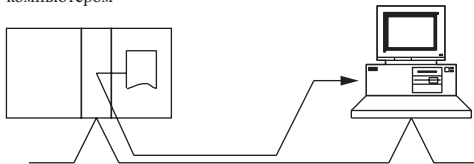
### Взаимодействие с другими устройствами в сети

Следующий ниже пример показывает взаимодействие Программируемого контроллера с устройствами в другом Программируемом контроллере (Модуль центрального процессора, Модуль шины центрального процессора, встроенная плата). Для ознакомления с деталями обратитесь к Руководству по работе в используемой сети (Controller Link или Ethernet).



Данный пример показывает взаимодействие Программируемого контроллера с персональным компьютером.

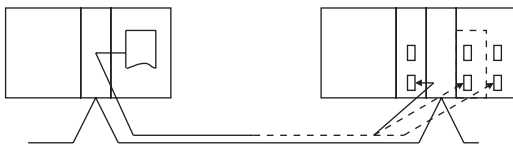
Взаимодействие Программируемого контроллера с персональным компьютером



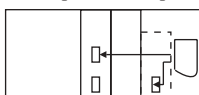
### Взаимодействие Программируемого контроллера с последовательными портами

Данные примеры показывают взаимодействие Программируемого контроллера с последовательными портами устройств, подключенных к сети. Первый пример показывает коммуникационный обмен с портами устройств, находящихся в другом Программируемом контроллере (Модуль центрального процессора, Модуль шины центрального процессора CS1, или Встроенная плата), а второй пример – коммуникационный обмен с последовательным портом, находящемся в самой Панели центрального процессора.

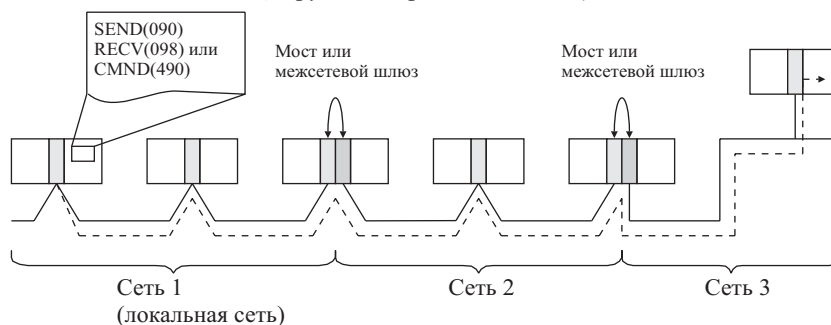
Взаимодействие в сети



Взаимодействие в Панели центрального процессора



**Примечание:** Коммуникационный обмен данными может осуществляться вплоть до 3-го сетевого уровня, включая локальную сеть. (Локальная сеть – это сеть, в которой иницируется передача данных.)



Для обмена через сеть, необходимо в каждом из Модулей центрального процессора Программируемого контроллера зарегистрировать таблицу маршрутизации, которая обозначает путь, по которому данные

будут передаваться в желаемый узел. Каждая из таблиц маршрутизации составляется из таблицы локальной сети и таблицы ретранслирующей сети.

- 1,2,3...**
1. *Таблица локальной сети.*  
Эта таблица показывает номера модулей и сетевые адреса узлов, соединенных с локальным Программируемым контроллером.
  2. *Таблица ретранслирующей сети.*  
Эта таблица показывает номера модулей и сетевые адреса первых ретранслирующих узлов, осуществляющих передачу данных в сети назначения, которые не соединены к локальному Программируемому контроллеру.

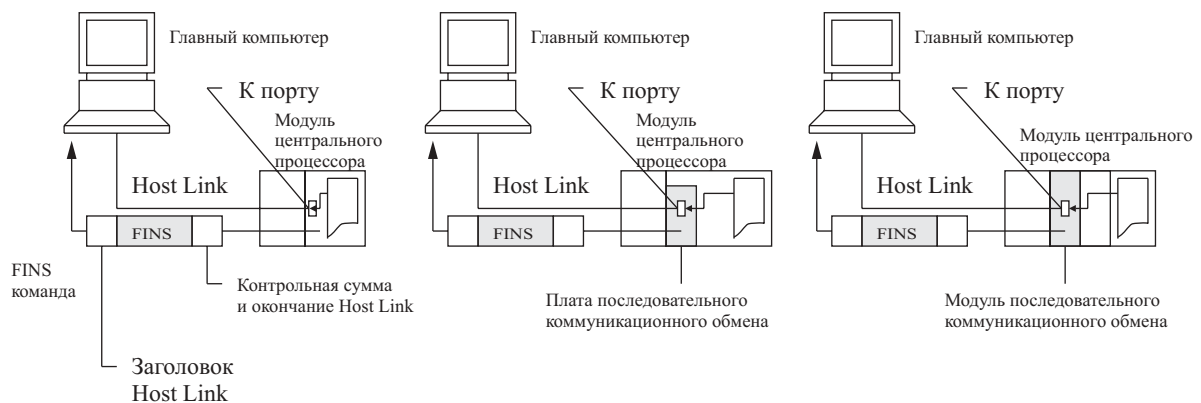
#### Взаимодействие с Главным компьютером (Host Link)

Посредством выдачи команд SEND(090), RECV(098) или CMND(490) последовательному порту, работающему в режиме Host Link, к FINS команде добавляется необходимый заголовок и окончание Host Link, и команда FINS посылается главному компьютеру.

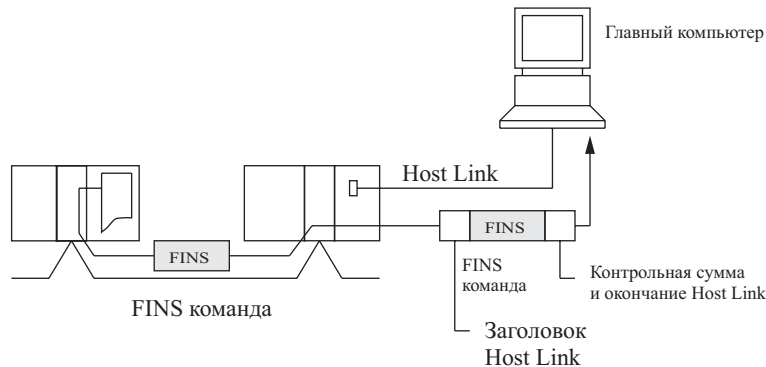
Главный компьютер, подключенный к встроенному порту Модуля центрального процессора

Главный компьютер, подключенный к Плате последовательного коммуникационного обмена

Главный компьютер, подключенный к Модулю последовательного коммуникационного обмена

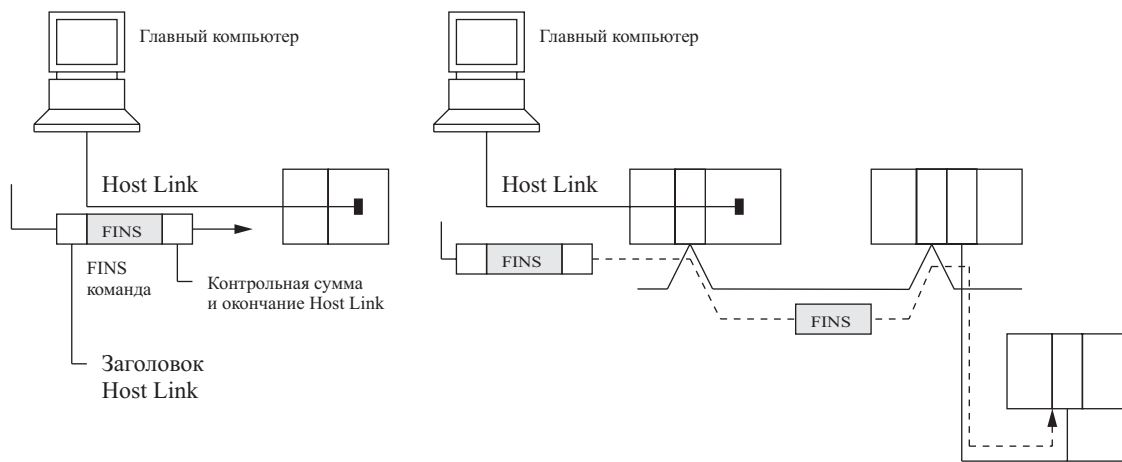


**Примечание:** Коммуникационный обмен Host Link может осуществляться через сеть. В этом случае прохождение команд FINS через сеть осуществляется нормально. Когда команда достигает системы Host Link, к FINS команде добавляется необходимый заголовок и окончание Host Link, и команда FINS посылается главному компьютеру.



#### Передача данных из главного компьютера

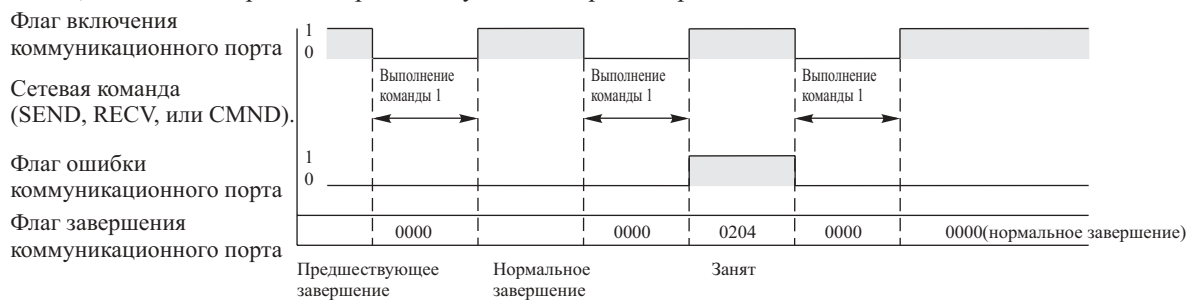
Команды FINS можно посылать из главного компьютера в подключенный к нему Программируемый контроллер, а также в другие устройства, подключенные к сети (Модули центрального процессора, Специальные модули, компьютеры и т.д.). В этом случае необходимый заголовок и окончание Host Link должны присоединяться к команде FINS при отправлении.



**Коммуникационные флаги**

Основные принципы функционирования коммуникационных флагов изложены ниже.

- Флаг ошибки коммуникационного порта обнуляется в процессе выполнения обмена, и устанавливается в значение, равное 1, когда обмен завершается (успешно или с ошибкой).
- Состояние Флага ошибки коммуникационного порта поддерживается до следующей передачи или приема данных.
- Флаг ошибки коммуникационного порта обнуляется в процессе следующей передачи или приема данных, даже если в процессе предшествующей операции произошла ошибка.

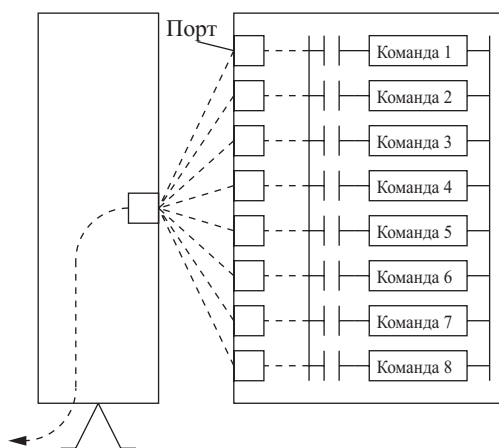


**О номерах коммуникационных портов**

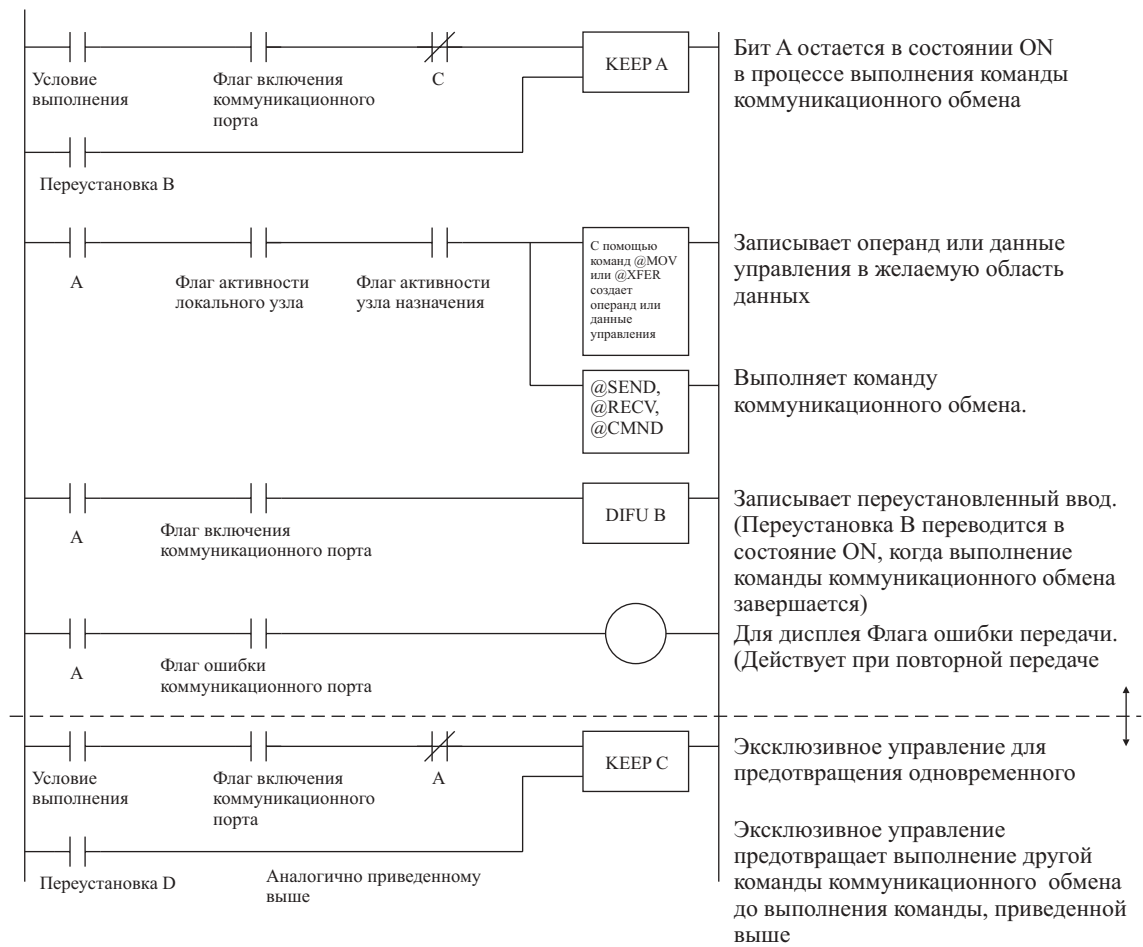
Существует 8 логических коммуникационных портов, поэтому одновременно может выполняться 8 коммуникационных команд. Один порт может выполнять только одну команду. В случае, когда выполняется более восьми команд, должно применяться эксклюзивное управление.

Эти 8 коммуникационных портов используются сетевыми командами (SEND(090), RECV(098), CMND(490)), и командой PROTOCOL MACRO (PMCR(260)). Убедитесь в том, что один порт не используется одновременно двумя командами.

Модуль Controller Link или Модуль Ethernet      Модуль центрального процессора



На следующем ниже рисунке приводится пример выполнения эксклюзивного управления.



### 3-23-2 Команда передачи данных при сетевом обмене NETWORK SEND: SEND(090)

#### Назначение

По команде SEND(090) осуществляется передача данных в узел сети.

#### Символ релейно-контактной схемы

SEND(090)	
S	S: Первое исходное слово (локальный узел)
D	D: Первое слово назначения (удаленный узел)
C	C: Первое слово управления

#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	SEND(090)
	Выполняется один раз для дифференцирования вверх	@ SEND(090)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****С: Первое слово управления**

Пять слов управления от С до С+4 указывают количество передаваемых слов, данные места назначения и другие установки, указанные в следующей ниже таблице.

Слово	Биты от 0-0 до 07	Биты от 08 до 15
С	Количество слов: разрешается от 0001 до максимального значения <sup>1</sup> (четырёхзначное шестнадцатеричное).	
С+1	Адрес сети назначения: от 00 до 7F (от 00 до 127) <sup>2</sup> .	Номер последовательного порта (от 01 до 04 шестн.) для коммуникационного обмена Host Link <sup>3</sup> .
С+2	Адрес модуля назначения: от 00 до FE <sup>4</sup> .	Адрес узла назначения: разрешается от 00 до максимального значения <sup>5</sup> .
С+3	Количество попыток: от 00 до 0F (от 0 до 15).	Биты 08 11: Номер коммуникационного порта (внутренний логический порт): от 0 до 7. Биты 12 15: Установки для отклика. 0: Отклик запрашивается. 8: Отклик не запрашивается <sup>6</sup> .
С+4	Время мониторинга отклика: 0001 FFFF (от 0.1 до 6553.5 секунд.) (Значение по умолчанию, равное 0000, задает время мониторинга, равное 2 сек.)	

- Примечание:**
1. Максимально допустимое количество слов зависит от используемой сети. Для сети Controller Link, допустимый диапазон значений – от 0001 до 03DE (1 990 слов).
  2. Для передачи данных в пределах локальной сети устанавливайте адрес сети назначения в значение 00. Когда подключается два и более Модулей шины центрального процессора CS1, адресом сети будет номер Модуля с наименьшим номером.
  3. Номера от 01 до 04 обозначают последовательные порты 1...4 (физические порты) устройства назначения.
  4. Адрес модуля указывает желаемый модуль, как показано в следующей ниже таблице.

Модуль	Установка адреса модуля
Модуль центрального процессора	00 шестн.
Модуль шины центрального процессора CS1	Номер модуля + 10 шестн.
Специальный модуль CS1	Номер модуля + 20 шестн.
Встроенная плата	E1 шестн.
Компьютер	01 шестн.
Модуль, подключенный к сети (указывать Модуль не обязательно)	FE шестн.

5. Максимальный номер узла зависит от типа используемой сети. Для сети Controller Link, диапазон допустимых значений – от 00 до 20 шестн. (0...32). Для передачи данных во все узлы устанавливайте номер узла назначения в значение FF. Для передачи данных в пределах локального узла задавайте значение 00.
6. Когда номер узла назначения устанавливается в значение FF (передача во все узлы), отклик не передается, даже если биты 12...15 установлены в значение, равное нулю.

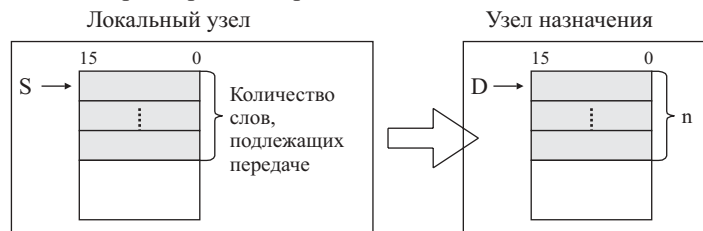
**Спецификации операндов**

Область	S	C	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6139	
Рабочая область	W000...W511	W000...W507	
Область удержания бита (Holding bit area)	H000...H511	H000...H507	
Область вспомогательных битов	A000...A447 A448...A959	A448...A959	A000...A443 A448...A955
Область таймера	T0000...T4095	T0000...T4091	
Область счетчика	C0000...C4095	C0000...C4091	
Область DM	D00000...D32767	D00000...D32763	
Область EM, не содержащая банков	E00000...E32767	E00000...E32763	

Область	S	C	D
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32763 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	–		
Регистры данных	–		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-(-)IR15		

### Описание

По команде SEND(090) осуществляется передача данных, начиная со слова S, в адреса устройства назначения, начиная со слова D. Передача производится через шину центрального процессора Программируемого контроллера или через сеть. Количество слов, подлежащих передаче, указывается в слове C.



Если номер узла назначения устанавливается в значение FF, данные передаются во все узлы сети назначения. Такая передача называется ширококвещательной передачей.

В случае, когда запрашивается отклик (биты 12...15 слова C+3 установлены в значение 0), а отклик не принимается в течение заданного времени мониторинга, производится повторная передача данных до 15 раз (количество повторений указывается в битах 0...3 слова C+3). При ширококвещательной передаче данных отклик не передается, и повторные попытки передачи данных не предпринимаются.

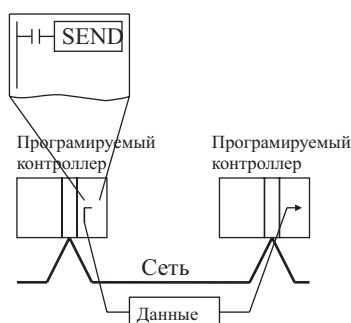
Команда SEND(090) может использоваться для передачи данных в отдельный последовательный порт устройства назначения и для передачи данных собственно в устройство назначения.

Данные могут передаваться в главный компьютер, подключенный к последовательному порту Программируемого контроллера (при установке режима Host Link), а также в Программируемый контроллер или компьютер, подключенный через сеть Controller Link или Ethernet.

Когда при выполнении команды SEND(090) флаг включения коммуникационного порта, указанного в C+3, находится в состоянии ON, Флаг включения соответствующего порта (порты 00...07: A20200...A20207) и Флаг ошибки коммуникационного порта (порты 00...07: A21900...A21907) переводятся в состояние OFF и в слово, содержащее код завершения, записывается значение 0000 (порты 0...7: A203...A210). Данные будут передаваться в узел назначения после установки флагов.

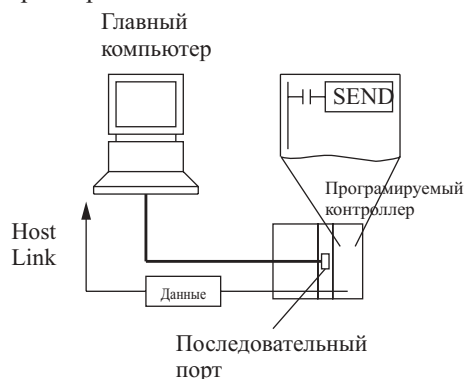
### Передача по сети

Команда SEND(090) может использоваться для передачи данных из Программируемого контроллера в указанную область данных в Программируемом контроллере или компьютере, подключенном через сеть Controller Link или Ethernet.



### Передача по соединению Host Link

Когда встроенный последовательный порт Модуля центрального процессора, Платы последовательного коммуникационного обмена или Модуль последовательного коммуникационного обмена работает в режиме Host Link и существует прямое соединение с главным компьютером, команда SEND(090) может выполняться для передачи данных из Программируемого контроллера в главный компьютер каждый раз, когда Программируемый контроллер получает право на передачу данных. Кроме того, можно производить передачу данных другим главным компьютерам, подключенным к другим Программируемым контроллерам в любой части сети.



Когда команда SEND(090) посылается последовательному порту Модуля центрального процессора, Платы последовательного коммуникационного обмена или Модуль последовательного коммуникационного обмена, команда пересылается из последовательного порта в главный компьютер. Команда является сообщением FINS, снабженным заголовком и окончанием Host Link. Командой FINS является команда MEMORY AREA WRITE (код команды 0102), а кодом заголовка Host Link является 0F (шестн.).

Для обработки принимаемых команд (команда FINS, снабженная заголовком и окончанием Host Link) в главном компьютере должна создаваться программа.

Если последовательный порт назначения находится в местном Программируемом контроллере, устанавливайте в слове C+1 адрес сети в значение 00 (локальная сеть), адрес узла в слове C+2 – в значение 00 (локальный Программируемый контроллер), и адрес модуля – в значение 00 (для Модуля центрального процессора), E1 (для Встроенной платы) или номер модуля +10 (шестн.) (для Модуля последовательного коммуникационного обмена).

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда номер последовательного порта, указанный в C+1 выходит за пределы значений от 00 до 04. Переводится в состояние ON, когда флаг включения указанного в C+3 коммуникационного порта находится в состоянии OFF. OFF в других случаях.

В следующей ниже таблице показаны относящиеся к команде биты и флаги вспомогательной области.



Наименование	Адрес	Содержание
Флаг включения коммуникационного порта	A20200...A20207	Эти флаги переводятся в состояние ON для индикации разрешения выполнения сетевых команд (включая команду PCMR(260)) для соответствующих портов (00...07). Флаг переводится в состояние OFF, когда для соответствующего порта выполняется сетевая команда. Флаг переводится в состояние ON, когда выполнение команды завершается.
Флаг ошибки коммуникационного порта	A21900...A21907	Переводится в состояние ON при возникновении на соответствующем порте (00...07) ошибки выполнения сетевой команды. Состояние флага поддерживается до начала выполнения следующей команды. Флаг переводится в состояние OFF, когда начинается следующий обмен данными, даже если при последнем выполнении произошла ошибка.
Коды завершения коммуникационного порта	A203...A210	Эти слова содержат коды завершения выполнения сетевой команды для соответствующего порта. В процессе осуществления сетевой команды соответствующий код окончания равен 0000. После завершения выполнения команды в это слово записывается код завершения. Содержание этих слов при запуске программы сбрасывается.

### Меры предосторожности

Если для указанного в C+3 порта флаг включения находится в состоянии OFF, команда принимается в качестве команды NOP(000) и, следовательно, не выполняется. В этом случае флаг ошибки переводится в состояние ON.

Когда для слова D указывается адрес в текущем банке области EM, передаваемые данные будут записаны в текущий банк EM области узла назначения.

Когда данные передаются за пределы локальной сети, пользователь обязан зарегистрировать таблицы маршрутизации в Программируемых контроллерах (модулях центрального процессора) в каждой сети. (Таблицы маршрутизации определяют пути к другим сетям, в которых находятся узлы назначения.)

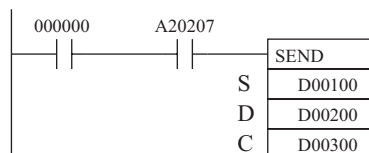
Только одна сетевая команда может выполняться портом одновременно. Для того чтобы команда SEND(090) не выполнялась в момент, когда порт занят, программируйте флаг включения коммуникационного порта (A20200...A20207) в качестве нормально открытого условия.

Коммуникационные порты 00...07 используются сетевыми командами и командой PCMR(260) совместно, поэтому команда SEND(090) не может выполняться одновременно с командой PCMR(260), если в обеих командах указывается один и тот же порт.

Вследствие помехи или других влияющих факторов, передаваемые данные или отклик могут быть повреждены или утеряны, поэтому рекомендуется устанавливать количество повторных попыток в значение, отличающееся от нуля. В этом случае команда SEND(090) будет выполняться повторно, если отклик не будет принят в течение установленного времени мониторинга.

### Примеры

В следующем ниже примере, когда CIO 000000 и A20207 (флаг включения коммуникационного порта 07) находятся в состоянии ON, десять слов, начиная с D00100 до D00109, передаются в узел 3 локальной сети, где они записываются в слова D00200...D00209. Данные будут передаваться повторно до трех раз, если отклик не принимается в течение десяти секунд.



C: D00300	0 0 0 A	Количество слов, подлежащих передаче
C+1: D00301	0 0 0 0	Передача в пределах локальной сети и в пределах устройства
C+2: D00302	0 3 0 0	Номер узла 3, адрес Модуля 00 (Модуль центрального процессора.)
C+3: D00303	0 7 0 3	Отклик запрашивается, порт 7, 3 повторные попытки
C+4: D00304	0 0 6 4	Время мониторинга отклика: 0064 (шестн.) (10 секунд)

## 3-23-3 Команда приема данных при сетевом обмене NETWORK RECEIVE: RECV(098)

**Назначение**

По команде RECV(098) осуществляется запрос данных, подлежащих передаче из узла сети, и прием этих данных.

**Символ релейно-контактной схемы**

—	RECV(098)	
	S	S: Первое исходное слово (удаленный узел)
	D	D: Первое слово назначения (локальный узел)
	C	C: Первое слово управления

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	RECV(098)
	Выполняется один раз для дифференцирования вверх	@ RECV(098)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****C: Первое слово управления**

Пять слов управления от C до C+4 указывают количество принимаемых слов, источник передачи данных и другие установки, указанные в следующей ниже таблице.

Слово	Биты от 0-0 до 07	Биты от 08 до 15
C	Количество слов: разрешается от 0001 до максимального значения <sup>1</sup> (четырёхзначное шестнадцатеричное).	
C+1	Адрес исходной сети: от 00 до 7F (от 00 до 127) <sup>2</sup> .	Номер последовательного порта (от 01 до 04 шестн.) для коммуникационного обмена Host Link <sup>3</sup> .
C+2	Адрес исходного модуля: от 00 до FE <sup>4</sup> .	Адрес исходного узла: разрешается от 00 до максимального значения <sup>5</sup> .
C+3	Количество попыток: от 00 до 0F (от 0 до 15).	Биты 08...11: Номер коммуникационного порта: от 00 до 07. Отклик обязательно запрашивается.
C+4	Время мониторинга отклика: 0001...FFFF (от 0.1 до 6553.5 секунд.) (Значение по умолчанию, равное 0000, задает время мониторинга, равное 2 сек.)	

- Примечание:**
1. Максимально допустимое количество слов зависит от используемой сети. Для сети Controller Link, допустимый диапазон значений – от 0001 до 03DE (1...990 слов).
  2. Для передачи данных в пределах локальной сети устанавливайте адрес исходной сети в значение 00. Когда используется два и более Модулей шины центрального процессора CS1, адресом сети будет номер Модуля с наименьшим номером.
  3. Номера от 01 до 04 обозначают последовательные порты 1...4 (внутренние логические порты) исходного устройства.
  4. Адрес модуля указывает желаемый модуль, как показано в следующей ниже таблице.

Модуль	Установка адреса модуля
Модуль центрального процессора	00 шестн.
Модуль шины центрального процессора CS1	Номер модуля + 10 шестн.
Специальный модуль CS1	Номер модуля + 20 шестн.
Встроенная плата	E1 шестн.
Компьютер	01 шестн.

Модуль	Установка адреса модуля
Модуль, подключенный к сети (указывать Модуль не обязательно)	FE шестн.

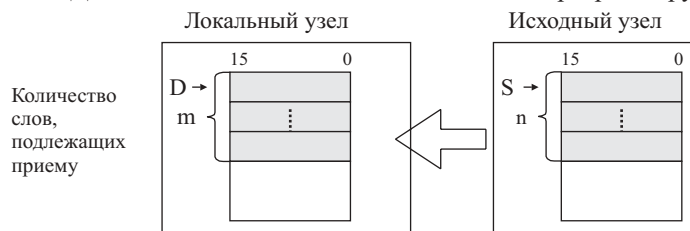
5. Максимальный номер узла зависит от типа используемой сети. Для сети Controller Link, диапазон допускаемых значений – от 00 до 20 шестн. (0...32). Для передачи данных во все узлы устанавливайте номер узла назначения в значении FF. Для передачи данных в пределах локального узла задавайте значение 00.

### Спецификации операндов

Область	S	C	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6139	
Рабочая область	W000...W511	W000...W507	
Область удержания бита (Holding bit area)	H000...H511	H000...H507	
Область вспомогательных битов	A000...A447 A448...A959	A448...A959	A000...A443 A448...A955
Область таймера	T0000...T4095	T0000...T4091	
Область счетчика	C0000...C4095	C0000...C4091	
Область DM	D00000...D32767	D00000...D32763	
Область EM, не содержащая банков	E00000...E32767	E00000...E32763	
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)	En_00000...En_32763 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	–		
Регистры данных	–		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-(-)IR15		

### Описание

По команде RECV(098) производится запрос указанного в C количества слов, начиная со слова S, для передачи данных из указанного устройства назначения в локальный Программируемый контроллер. Прием данных производится через шину центрального процессора Программируемого контроллера или через сеть. Данные записываются в область данных Программируемого контроллера, начиная с D.



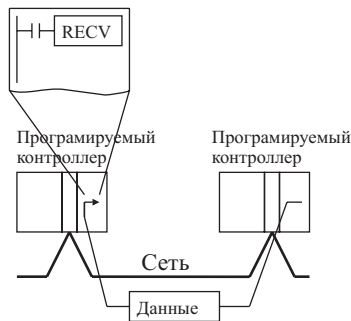
Отклик запрашивается командой RECV(098), так как отклик содержит сведения о принятых данных. Если отклик не принимается в течение заданного в C+4 времени мониторинга, запрос на передачу данных будет повторяться до 15 раз (количество повторов устанавливается в битах 0...3 слова C+3).

Команда RECV(098) может использоваться для запроса передачи данных через определенный последовательный порт исходного устройства или собственно из устройства.

Данные могут приниматься из главного компьютера, подключенного к последовательному порту Программируемого контроллера (при установке режима Host Link), а также из Программируемого контроллера или компьютера, подключенного через сеть Controller Link или Ethernet.

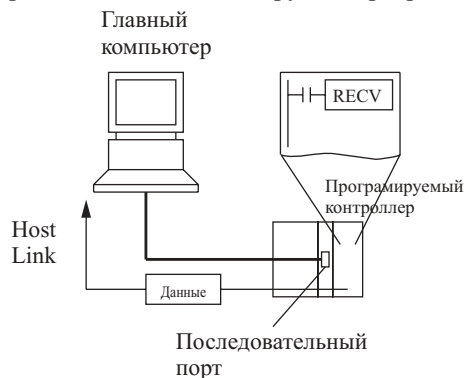
### Передача по сети

Команда RECV(098) может использоваться для приема данных из указанной области данных Программируемого контроллера или компьютера, подключенного через сеть Controller Link или Ethernet. Принятые данные записываются в указанную область данных локального Программируемого контроллера.



### Передача по соединению Host Link

Когда последовательный порт Модуля центрального процессора, Платы последовательного коммуникационного обмена или Модуль последовательного коммуникационного обмена работает в режиме Host Link и существует прямое соединение с главным компьютером, команда RECV(098) может выполняться для приема данных из главного компьютера каждый раз, когда Программируемый контроллер получает право на передачу данных. Кроме того, можно производить прием данных из других главных компьютеров, подключенных к другим Программируемым контроллерам в любой части сети.



Когда команда RECV(098) выполняется для последовательного порта Модуля центрального процессора, Платы последовательного коммуникационного обмена или Модуля последовательного коммуникационного обмена, команда пересылается из последовательного порта в главный компьютер. Команда является сообщением FINS, снабженным заголовком и окончанием Host Link. Командой FINS является команда MEMORY AREA READ (код команды 0101), а кодом заголовка Host Link является 0F (шестн.).

Для обработки передаваемых команд (команда FINS, снабженная заголовком и окончанием Host Link) в главном компьютере должна создаваться программа.

Если последовательный порт назначения находится в локальном Программируемом контроллере, устанавливайте в слове C+1 адрес сети в значение 00 (локальная сеть), адрес узла в слове C+2 – в значение 00 (локальный Программируемый контроллер), и адрес модуля – в значение 00 (для Модуля центрального процессора), E1 (для Встроенной платы) или номер модуля +10 (шестн.) (для Модуля последовательного коммуникационного обмена).

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда номер последовательного порта, указанный в C+1 выходит за пределы значений от 00 до 04. Переводится в состояние ON, когда флаг включения указанного в C+3 коммуникационного порта находится в состоянии OFF. OFF в других случаях.

В следующей ниже таблице показаны относящиеся к команде биты и флаги вспомогательной области.

Наименование	Адрес	Содержание
Флаг включения коммуникационного порта	A20200...A20207	Эти флаги переводятся в состояние ON для индикации разрешения выполнения сетевых команд (включая команду PCMR(260)) для соответствующих портов (00...07). Флаг переводится в состояние OFF, когда для соответствующего порта выполняется сетевая команда. Флаг переводится в состояние ON, когда выполнение команды завершается.
Флаг ошибки коммуникационного порта	A21900...A21907	Переводится в состояние ON при возникновении на соответствующем порте (00...07) ошибки выполнения сетевой команды. Состояние флага поддерживается до начала выполнения следующей команды. Флаг переводится в состояние OFF, когда начинается следующий обмен данными, даже если при последнем выполнении произошла ошибка.
Коды завершения коммуникационного порта	A203...A210	Эти слова содержат коды завершения выполнения сетевой команды для соответствующего порта. В процессе осуществления сетевой команды соответствующий код окончания равен 0000. После завершения выполнения команды в это слово записывается код завершения. Содержание этих слов при запуске программы сбрасывается.

### Меры предосторожности

Только одна сетевая команда может выполняться портом одновременно. Для того, чтобы команда RECV(098) не выполнялась в момент, когда порт занят, программируйте флаг включения коммуникационного порта (A20200...A20207) в качестве нормально открытого условия.

Коммуникационные порты 00...07 используются сетевыми командами и командой PCMR(260) совместно, поэтому команда RECV(098) не может выполняться одновременно с командой PCMR(260), если в обеих командах указывается один и тот же порт.

Вследствие помехи или других влияющих факторов, передаваемые данные или отклик могут быть повреждены или утеряны, поэтому рекомендуется устанавливать количество повторных попыток в значение, отличающееся от нуля. В этом случае команда RECV(098) будет выполняться повторно, если отклик не будет принят в течение установленного времени мониторинга.

### 3-23-4 Передача команды DELIVER COMMAND: CMND(490)

#### Назначение

По команде CMND(490) осуществляется передача FINS команд и прием отклика.

#### Символ релейно-контактной схемы

— CMND(490)	
S	S: Первое слово команды
D	D: Первое слово отклика
C	C: Первое слово управления

#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	CMND(490)
	Выполняется один раз для дифференцирования вверх	@ CMND(490)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

##### С: Первое слово управления

Шесть слов управления от С до С+ 5 указывают количество данных команды и данных отклика, данные места назначения и другие установки, указанные в следующей ниже таблице.

Слово	Биты от 0-0 до 07	Биты от 08 до 15
C	Количество байтов данных в команде: разрешается от 0002 до максимального значения <sup>1</sup> (четырёхзначное шестнадцатеричное).	
C+1	Количество байтов данных в отклике: разрешается от 0000 до максимального значения <sup>1,2</sup> (четырёхзначное шестнадцатеричное).	
C+2	Адрес сети назначения: от 00 до 07 <sup>3</sup> .	00 или номер последовательного порта (от 01 до 04) <sup>4</sup> .
C+3	Адрес модуля назначения: от 00 до FE <sup>5</sup> .	Адрес узла назначения: разрешается от 00 до максимального значения <sup>6</sup> .
C+4	Количество попыток: от 00 до 0F (от 0 до 15).	Биты 08...11: Номер коммуникационного порта (внутренний логический порт): от 0 до 7. Биты 12...15: Установки для отклика. 0: Отклик запрашивается 8: Отклик не запрашивается <sup>7</sup> .
C+5	Время мониторинга отклика: 0001...FFFF (от 0.1 до 6553.5 секунд.) (Значение по умолчанию, равное 0000, задает время мониторинга, равное 2 сек.)	

**Примечание:**

1. Максимально допустимое количество байтов в данных команды и отклика зависит от используемой сети. Для сети Controller Link, максимальное значение – 07C6 (1990 байтов).
2. Если действительное количество байтов отклика превышает заданное в C+1 значение, данные отклика не запоминаются. Если действительное количество байтов отклика меньше указанного в C+1, принятые данные записываются, а остаток слов отбрасывается, т.к. отклик должен записываться в том виде, в котором он принят.
3. Устанавливайте адрес сети назначения в значение 00 для передачи данных в пределах локальной сети. Когда устанавливается два и более Модулей шины центрального процессора CS1, адресом сети будет номер Модуля с наименьшим номером.
4. При передаче FINS команд главному компьютеру через сеть Host Link, указывайте номер последовательного порта. Номера от 01 до 04 обозначают последовательные порты 1...4 (физические порты) устройства назначения.
5. Адрес модуля указывает желаемый модуль, как показано в следующей ниже таблице.

Модуль	Установка адреса модуля
Модуль центрального процессора	00 шестн.
Модуль шины центрального процессора CS1	Номер модуля + 10 шестн.
Специальный модуль CS1	Номер модуля + 20 шестн.
Встроенная плата	E1 шестн.
Компьютер	01 шестн.
Модуль, подключенный к сети (указывать Модуль не обязательно)	FE шестн.

6. Максимальный номер узла зависит от типа используемой сети. Для сети Controller Link, диапазон допускаемых значений – от 00 до 20 шестн. (0...32). Для передачи данных во все узлы устанавливайте номер узла назначения в значение FF. Для передачи данных в пределах локального узла задавайте значение 00.
7. Когда номер узла назначения устанавливается в значение FF (широковещательная передача), отклик не передается, даже если биты 12...15 установлены в значение, равное нулю.

**Спецификации операндов**

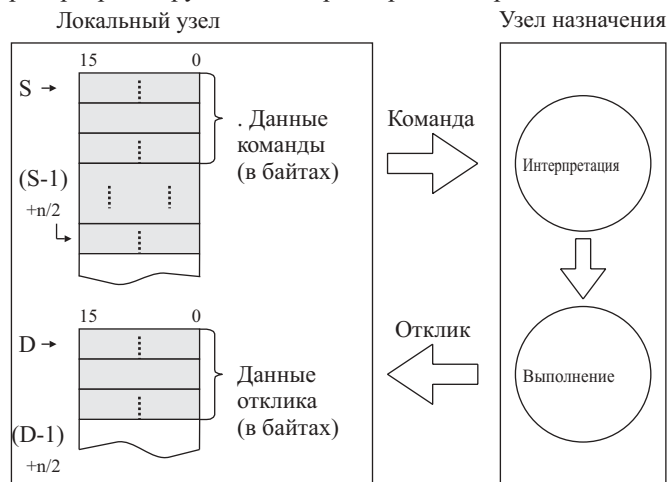
Область	S	C	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		CIO 0000...CIO 6138
Рабочая область	W000...W511		W000...W506
Область удержания бита (Holding bit area)	H000...H511		H000...H506
Область вспомогательных битов	A000...A447 A448...A959	A448...A959	A000...A442 A448...A954
Область таймера	T0000...T4095		T0000...T4090
Область счетчика	C0000...C4095		C0000...C4090



Область	S	C	D
Область DM	D00000...D32767		D00000...D32762
Область EM, не содержащая банков	E00000...E32767		E00000...E32762
Область EM, содержащая банки	En_00000...En_32767 (n = 0...C)		En_00000...En_32763 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767		
Константы	-		
Регистры данных	-		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-( -)IR15		

### Описание

По команде CMND(490) осуществляется передача указанного количества байтов данных FINS команды, начиная со слова S, в устройство назначения. Передача производится через шину центрального процессора Программируемого контроллера или через сеть. Отклик записывается в память, начиная со слова D.



Команда CMND(490) может применяться для передачи данных команды в указанный порт устройства назначения, а также собственно в устройство назначения. Команда CMND(490) функционирует подобно команде SEND(090), если кодом команды FINS является 0102 (MEMORY AREA WRITE) или подобно команде RECV(098), если кодом команды FINS является 0101 (MEMORY AREA READ).

Модуль центрального процессора, выполняющий команду CMND(490), может посылать FINS команду самому себе (только -EV1). Для выполнения такой операции используйте следующие ниже установи данных управления.

- Адрес сети назначения (биты 00...07 слова C+2): 00 (шестн.) (локальная сеть).
- Номер последовательного порта (биты 08 11 слова C+2): 0 (шестн.) (не используется).
- Адрес модуля назначения (биты 00...07 слова C+3): 00 (шестн.) (Модуль центрального процессора).
- Адрес узла назначения (биты 08...15 слова C+3): 00 (шестн.) (локальный узел).
- Количество повторных попыток (биты 00...03 слова C+4): 0 (шестн.) (данная функция не выполняется; задавайте значение 0).
- Время мониторинга отклика (биты 00...15 слова C+5): 0000...FFFF шестн. (однако в данном случае 0000 соответствует значению 6553.5 сек., в отличие от обычного случая, когда данному значению соответствует 2 сек.).

Если номер узла назначения устанавливается в значение FF, данные передаются во все узлы сети назначения. Такая передача называется ширококвещательной передачей.

В случае, когда запрашивается отклик (биты 12...15 слова C+3 установлены в значение 0), а отклик не принимается в течение заданного времени мониторинга, производится повторная передача данных до 15

раз (количество повторений указывается в битах 0...3 слова C+3). При широковещательной передаче данных отклик не передается, и повторные попытки передачи данных не предпринимаются. Для команд, которые не требуют получения отклика, устанавливайте соответствующее значение.

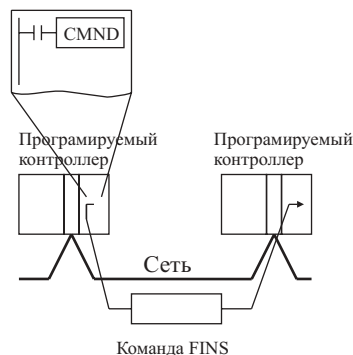
Если количество данных отклика превышает указанное в слове C+1, определяется ошибка.

Данные команды FINS могут передаваться в главный компьютер, подключенный к последовательному порту Программируемого контроллера (при установке режима Host Link), а также в Программируемый контроллер (Модуль центрального процессора, Модуль шины центрального процессора или Встроенную плату) или компьютер, подключенный через сеть Controller Link или Ethernet.

Когда при выполнении команды CMND(090) флаг включения коммуникационного порта, указанного в C+3, находится в состоянии ON, Флаг включения соответствующего порта (порты 00...07: A20200...A20207) и Флаг ошибки коммуникационного порта (порты 00...07: A21900...A21907) переводятся в состояние OFF и в слово, содержащее код завершения записывается значение 0000 (порты 0...7: A203...A210). Данные команды будут передаваться в узел (узлы) назначения один раз при установке флага.

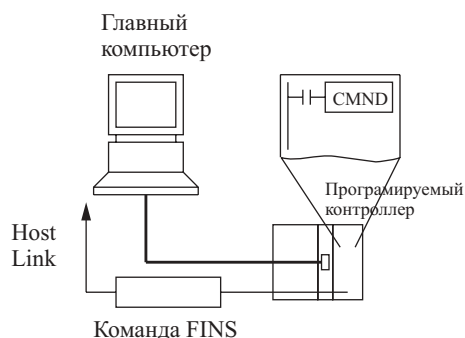
### Передача по сети

Команда CMND(490) может использоваться для передачи любой FINS команды в персональный компьютер или Программируемый контроллер (Модуль центрального процессора, Модуль шины центрального процессора или Встроенную плату), подключенный через сеть Controller Link или Ethernet.



### Передача по соединению Host Link

Когда последовательный порт Модуля центрального процессора, Платы последовательного коммуникационного обмена или Модуль последовательного коммуникационного обмена работает в режиме Host Link и существует прямое соединение с главным компьютером, команда CMND(490) может выполняться для передачи любой FINS команды из Программируемого контроллера в главный компьютер каждый раз, когда Программируемый контроллер получает право на передачу. Кроме того, можно производить передачу другим главным компьютерам, подключенным к другим Программируемым контроллерам в любой части сети.



Команда CMND(490) может выполняться для порта Модуля центрального процессора, Платы последовательного коммуникационного обмена или Модуля последовательного коммуникационного обмена с целью передачи команды в подключенный главный компьютер. (Задавайте номер последовательного порта в виде 1 или 2 (шестн.) в битах 08...11 слова C+2.) Команда является сообщением FINS, снабженным заголовком и окончанием Host Link. Можно передавать любую из FINS команд. Кодом заголовка Host Link является 0F (шестн.).

Для обработки принимаемых команд (команда FINS, снабженная заголовком и окончанием Host Link) в главном компьютере должна создаваться программа.

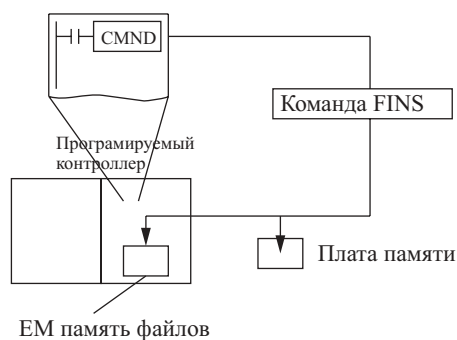


Если последовательный порт назначения находится в местном Программируемом контроллере, устанавливайте в слове С+2 адрес сети в значение 00 (локальная сеть), адрес узла в слове С+3 – в значение 00 (локальный Программируемый контроллер), и адрес модуля – в значение 00 (для Модуля центрального процессора), Е1 (для Встроенной платы) или номер модуля +10 (шестн.) (для Модуля последовательного коммуникационного обмена).

#### Передача FINS команды Модулю центрального процессора, выполняя команду CMND(490) (только модели EV1)

Модуль центрального процессора, выполняющий команду CMND(490), может передавать команду FINS самому себе (только модели EV1). Например, команды обработки файлов памяти (коды команд 22 шестн.) могут пересылаться для выполнения форматирования памяти файлов, удаления файлов, копирования файлов, и выполнения других операций. Для детального ознакомления обратитесь к Главе 12 «Функции памяти файлов» в Руководстве по эксплуатации Модуля центрального процессора CS1.

Флаг работы памяти файлов (A34313) переводится в состояние ON, когда в местный Модуль центрального процессора передается любая из FINS команд (даже FINS команда, не имеющая отношения к памяти файлов). Поэтому всегда используйте флаг A34313 в качестве нормально замкнутого условия ввода команды CMND(490) с целью предотвращения одновременного выполнения более чем одной команды FINS.



#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда номер последовательного порта, указанный в С+2 выходит за пределы значений от 00 до 04. Переводится в состояние ON, когда флаг включения указанного в С+4 коммуникационного порта находится в состоянии OFF. Переводится в состояние ON, если FINS команда передается в местный Программируемый контроллер, когда флаг работы памяти файлов (A34313) находится в состоянии ON. OFF в других случаях.

В следующей ниже таблице показаны относящиеся к команде биты и флаги вспомогательной области.

Наименование	Адрес	Содержание
Флаг включения коммуникационного порта	A20200...A20207	Эти флаги переводятся в состояние ON для индикации разрешения выполнения сетевых команд (включая команду PMCR(260)) для соответствующих портов (00...07). Флаг переводится в состояние OFF, когда для соответствующего порта выполняется сетевая команда. Флаг переводится в состояние ON, когда выполнение команды завершается.
Флаг ошибки коммуникационного порта	A21900...A21907	Переводится в состояние ON при возникновении на соответствующем порте (00...07) ошибки выполнения сетевой команды. Состояние флага поддерживается до начала выполнения следующей команды. Флаг переводится в состояние OFF, когда начинается выполнение следующей команды, даже если при последнем выполнении произошла ошибка.
Коды завершения коммуникационного порта	A203...A210	Эти слова содержат коды завершения выполнения сетевой команды для соответствующего порта. В процессе выполнения сетевой команды соответствующий код окончания равен 0000. После завершения выполнения команды в это слово записывается код завершения. Содержание этих слов при запуске программы сбрасывается.

Наименование	Адрес	Содержание
Флаг работы памяти файлов	A34313	Переводится в состояние ON при передаче любой из FINS команд местному Модулю центрального процессора (даже если команда не имеет отношения к памяти файлов), или когда любая из следующих команд или операций выполняется с памятью файлов. FREAD(700), FWRIT(701) Перезапись программы при помощи бита управления и операция создания резервной копии.

### Меры предосторожности

Если для указанного в С+4 порта флаг включения находится в состоянии OFF, команда принимается в качестве команды NOP(000) и, следовательно, не выполняется. В этом случае флаг ошибки переводится в состояние ON.

Когда данные передаются за пределы локальной сети, пользователь обязан зарегистрировать таблицы маршрутизации в Программируемых контроллерах (Модулях центрального процессора) в каждой сети. (Таблицы маршрутизации определяют пути к другим сетям, в которых находятся узлы назначения.)

Только одна сетевая команда может выполняться портом одновременно. Для того чтобы команда CMND(490) не выполнялась в момент, когда порт занят, программируйте флаг включения коммуникационного порта (A20200...A20207) в качестве нормально открытого условия.

Коммуникационные порты 00...07 используются сетевыми командами и командой PCMR(260) совместно, поэтому команда CMND(490) не может выполняться одновременно с командой PCMR(260), если в обеих командах указывается один и тот же порт.

Всегда используйте один из флагов включения коммуникационного порта (A20200 – A20207) в нормально разомкнутых условиях ввода, а флаг работы памяти файлов (A34313) в нормально замкнутых условиях ввода команды CMND(490) при передаче FINS команды в местный Модуль центрального процессора.

Вследствие помехи или других влияющих факторов, передаваемые данные или отклик могут быть повреждены или утеряны, поэтому рекомендуется устанавливать количество повторных попыток в значение, отличающееся от нуля. В этом случае команда CMND(490) будет выполняться повторно, если отклик не будет принят в течение установленного времени мониторинга.

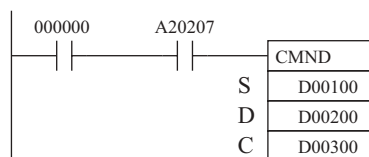
### Примеры

Следующий ниже раздел программы приводится в качестве примера передачи FINS команды удаленному Модулю центрального процессора.

Когда CIO 000000 и A20207 (флаг включения коммуникационного порта 07) находятся в состоянии ON, по команде CMND(490) в узел 3 передается FINS команда 0101 (MEMORY AREA READ). Отклик записывается в слова D00200...D00209.

По команде MEMORY AREA READ осуществляется чтение десяти слов, начиная с D00100 до D00109. Отклик содержит код команды длиной 2 байта (0101), код завершения длиной 2 байта, и 10 слов данных, что составляет всего 12 слов или 24 байта.

Данные будут передаваться повторно до трех раз, если отклик не принимается в течение десяти секунд.



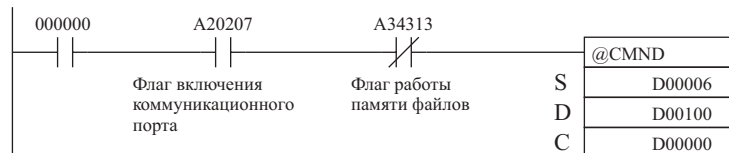
S: D00100	15 8 7 0	0 1 0 1	} D00010 (Область данных= 82 шестн. Адрес = 000A00)
S+1:D00101	8 2 0 0		
S+2:D00102	0 A 0 0		
S+3:D00103	0 0 0 A		
Количество слов, подлежащих чтению= 0A шестн. (10 в десятичном коде)			

C: D00300	15 8 7 0	0 0 0 8	Байты данных команды: 0008 (8 в десятичном коде)
C+1:D00301	0 0 1 8		Байты данных отклика: 0018 (24).
C+2:D00302	0 0 0 0		Передача в локальную сеть и собственно в устройство
C+3:D00303	0 3 0 0		Узел номер 3, адрес модуля 00 (Модуль центрального процессора).
C+3:D00304	0 7 0 3		Отклик запрашивается, порт 7, 3 повторные попытки
C+3:D00305	0 0 6 4		Время мониторинга отклика: 0064 (шестн.) (10 сек.).

Следующий ниже раздел программы показывает пример передачи FINS команды в местный Модуль центрального процессора.

Когда CIO 000000 и A20207 (флаг включения коммуникационного порта 07) находятся в состоянии ON, а A34313 (Флаг работы памяти файлов) - в состоянии OFF, по команде CMND(490) в местный Модуль центрального процессора передается FINS команда 2215 (CREATE/DELETE DIRECTORY).

Отклик записывается в слова D00100...D00101. В данном случае по команде FINS создается каталог, называемый CS1, в каталоге OMRON. Код команды (2 байта) и код окончания (2 байта) возвращаются и запоминаются в качестве отклика.



	15	8	7	0	
S: D00006	2	2	1	5	Код команды: 2215 шестн. (CREATE/DELETE DIRECTORY). Номер диска: 8000 шестн. (Плата памяти). Параметр 0000 шестн. (создание каталога).
S+1:D00007	8	0	0	0	
S+2:D00008	0	0	0	0	
S+3:D00009	4	3	5	3	Наименование каталога: CS1 □□□□□.□□□ (□ = пробел)
S+4:D00010	3	1	2	0	
S+5:D00011	2	0	2	0	
S+6:D00012	2	0	2	0	
S+7:D00013	2	E	2	0	
S+8:D00014	2	0	2	0	
S+9:D00015	0	0	0	6	Длина имени каталога: 0006 (6 символов).
S+10:D00016	5	C	4	F	Путь к основному каталогу: \OMRON
S+11:D00017	4	D	5	2	
S+12:D00018	4	F	4	E	

	15	8	7	0	
S: D00000	0	0	1	A	Байты данных команды: 001A (26 в десятичном коде).
S+1:D00001	0	0	0	4	Байты данных отклика: 0004 (4).
S+2:D00002	0	0	0	0	Адрес сети назначения: 00 шестн. (локальная сеть).
S+3:D00003	0	0	0	0	Адрес модуля назначения: 00 шестн. Номер узла назначения: 00 шестн. (Модуль центрального процессора в локальном узле)
S+4:D00004	0	7	0	0	Отклик запрашивается, порт 7, 0 повторных попыток.
S+5:D00005	0	0	0	0	Время мониторинга отклика: 0000 (шестн.) (6553.5 сек.).

## 3-24 Команды выполнения операций с памятью файлов

Настоящий раздел описывает команды, используемые для выполнения операций с памятью файлов (область EM или Платы памяти.)

**Примечание:** Память файлов может подвергаться изменениям посредством выполнения команды CMND(490) для передачи FINS команды в местный Модуль центрального процессора. Для детального ознакомления обратитесь к *Руководству по эксплуатации Программируемых контроллеров серии CS1*.

Команда	Мнемоническое обозначение	Функциональный код	Страница
READ DATA FILE	FREAD	700	621
WRITE DATA FILE	FWRIT	701	627

## 3-24-1 Команда чтения данных файла READ DATA FILE: FREAD(700)

**Назначение**

По команде FREAD(700) осуществляется чтение указанных данных или некоторого количества данных в указанном файле памяти файлов и запись данных в указанную область данных Модуля центрального процессора.

**Символ релейно-контактной схемы**

FREAD(700)	
C	C: Слово управления
S1	S1: Количество слов и первое исходное слово
S2	S2: Имя файла
D	D: Первое слово назначения

**Модификации**

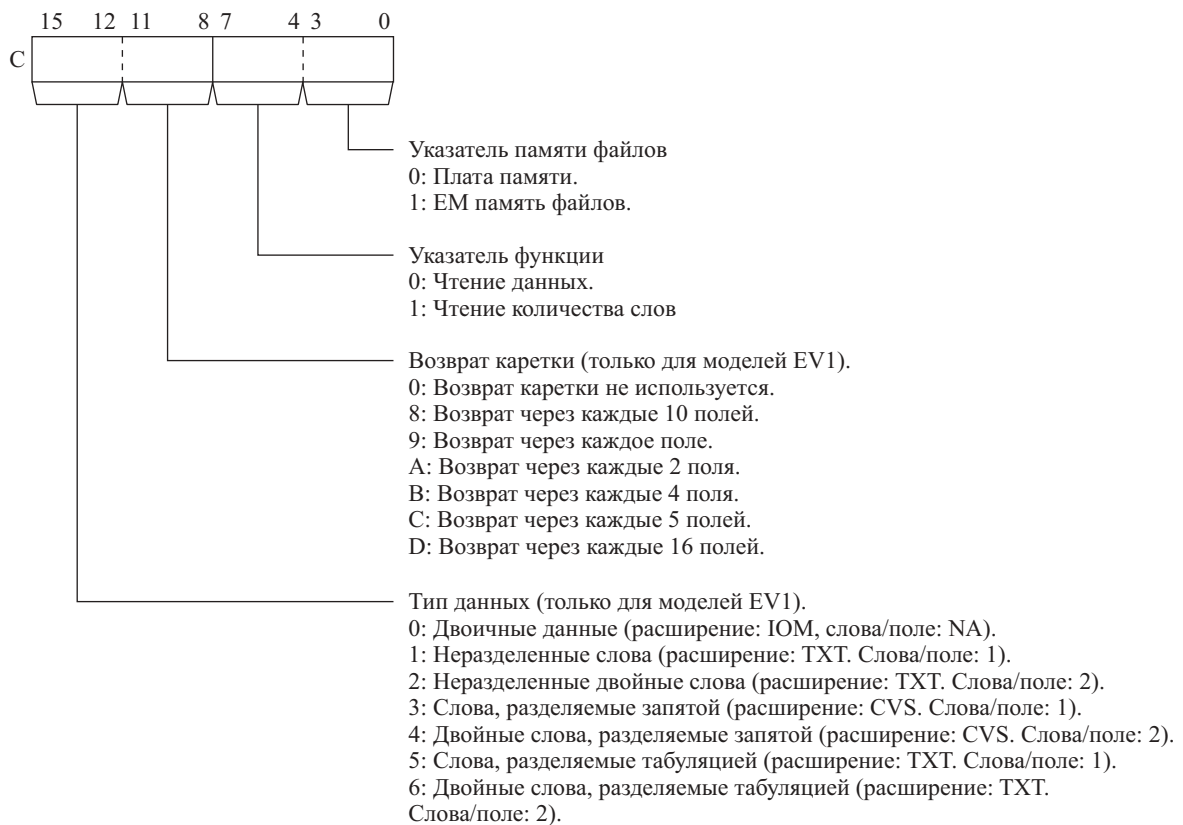
Модификации	Выполняется в каждом цикле для состояния ON	FREAD(700)
	Выполняется один раз для дифференцирования вверх	@ FREAD(700)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****C: Слово управления**

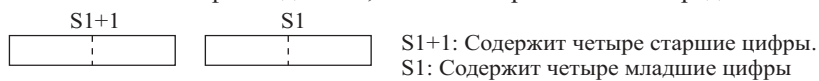
Как показано на следующем ниже рисунке, первая цифра обозначает место нахождения исходного файла – в Плате памяти или в EM памяти файлов. Вторая цифра слова управления указывает, будут ли читаться реальные данные или только некоторое количество слов данных. Третья цифра указывает наличие возврата каретки, а четвертая – тип данных.



- Примечание:**
1. Каждое поле содержит одно слово памяти ввода/вывода для данных длиной в одно слово, и два слова памяти ввода/вывода для данных длиной два слова.
  2. При чтении данных с использованием возврата каретки, биты 00...11 слова С должны быть установлены в значение между 8 и D (шестн.).
  3. При работе с двойными словами первое слово данных сохраняется в адресе памяти, имеющем более высокий номер, т.е. 12345678 будет запоминаться следующим образом: 1234 в слове D00001, а 5678 – в слове D00000.

#### S1 и S1+1: Количество читаемых элементов

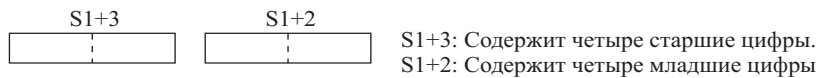
Восьмизначное шестнадцатеричное число в словах S+1 и S1+1 указывает количество слов или полей памяти файлов, подлежащих чтению. Если указанное количество слов или полей превышает количество слов или полей в файле данных, чтение завершается без определения ошибки.



Тип данных	Биты 12 15	Содержание S1 и S1+1
Двоичные данные	0 (шестн.) (двоичные)	Количество слов, подлежащих чтению в памяти файлов. 00000000...3FFFFFFF (шестн.)
Слово	1 (шестн.) (неразделенные) 2 (шестн.) (разделенные запятой) или 5 (шестн.) (разделенные табуляцией)	Количество полей, подлежащих чтению в памяти файлов, т.е. количество слов, подлежащих чтению в памяти файлов. 00000000...1FFFFFFF (шестн.)
Двойное слово	2 (шестн.) (неразделенные) 4 (шестн.) (разделенные запятой) или 6 (шестн.) (разделенные табуляцией)	Количество полей, подлежащих чтению в памяти файлов, т.е. половина количества слов, подлежащих чтению в памяти файлов. 00000000...0FFFFFFF (шестн.)

#### S1+2 и S1+3: Первое исходное слово

Восьмизначное шестнадцатеричное число в словах S1+2 и S1+3 указывает первое слово от начала файла, с которого начинается чтение.



Тип данных	Биты 12 15	Содержание S1+3 и S1+2
Двоичные данные	0 (шестн.) (двоичные)	Слово, отстоящее от начала файла, с которого начинается чтение данных. 00000000...3FFFFFFF (шестн.)
Слово	1 (шестн.) (неразделенные) 3 (шестн.) (разделенные запятой) 5 (шестн.) (разделенные табуляцией)	Поле, отстоящее от начала файла, с которого начинается чтение данных, т.е. количество слов, от начала файла. 00000000...1FFFFFFF (шестн.)
Двойное слово	2 (шестн.) (неразделенные) 4 (шестн.) (разделенные запятой) 6 (шестн.) (разделенные табуляцией)	Поле, отстоящее от начала файла, с которого начинается чтение данных, т.е. половина количества слов, от начала файла. 00000000...0FFFFFFF (шестн.)

- Примечание:**
- Слова S1+2 и S1+3 используются только для чтения текста и данных CVS без возврата каретки (т.е. биты 08...11 слова C устанавливаются в значение, равное 0) или для чтения двоичных данных. Всегда устанавливайте S1+2 и S1+3 в значение 00000000 (шестн.) при чтении данных с использованием возврата каретки (т.е. биты 08...11 слова C устанавливаются в значение между 8 и D (шестн.)).
  - Слова S1...S1+3 должны находиться в одной области данных.
  - Слова S1...S1+3 используются только для чтения данных.
  - Если указываемое первое слово превышает количество слов или полей в файле данных, флаг ошибки чтения файла (A34310) переводится в состояние ON, и чтение файла данных не производится.

### S2: Имя файла

S2 является начальным адресом слов, содержащих полное путевое имя и имя файла в кодах ASCII. Используйте коды ASCII от а до z, от А до Z, и от 0 до 9.

Полный путь к каталогу, содержащему файл данных, может содержать до 65 знаков, включая наклонную черту (ASCII 5C). Имя файла может быть длиной до 8 символов, однако имя не должно содержать нулей (ASCII 00), так как ноль используется для обозначения конца символьной строки. Не включайте расширение в имя файла; расширение .IOM добавляется автоматически.

S2	F1	F2	Содержат символьную строку, начиная со старшего байта слова S2 Полный путь к каталогу и имя файла может состоять из 74 символов (байтов), включая наклонную черту и ноль, указывающий конец строки
S2+1	F3	F4	
⋮	⋮	⋮	
S2+38	F73	F74	

- Примечание:**
- Убедитесь в том, что символьная строка, содержащая путь к каталогу и имя файла, не выходит за пределы области данных.
  - Если указываемый файл или каталог не существует, флаг безрезультатного обращения к файлу (A34311) переводится в состояние ON, и чтение данных не выполняется.

Записывайте путь и имя файла в кодах ASCII, начиная со старшего байта слова S2, как показано в следующем ниже примере для \ABC\XYZ.IOM. (Расширение .IOM добавляется автоматически.)

S2	"/"	"A"	S2	5C	41
S2+1	"B"	"C"	S2+1	42	43
S2+2	"/"	"X"	S2+2	5C	58
S2+3	"Y"	"Z"	S2+3	59	5A
S2+4	NUL		S2+4	00	

### D: Первое слово назначения

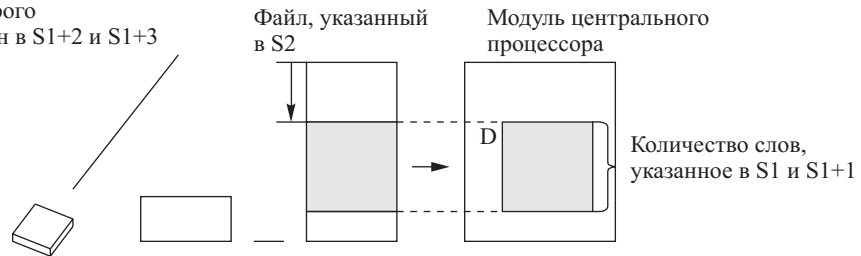
При выполнении чтения данных слово D указывает начальный адрес, куда будут записаны прочитанные данные.

Когда производится чтение некоторого количества слов данных, количество слов записывается в D и D+1 в виде восьмизначного шестнадцатеричного числа (00000000...FFFFFFF). D содержит четыре старшие цифры, а D+1 – четыре младшие цифры.

**Описание****Чтение данных (третья цифра слова С равна 0)**

По команде FREAD(700) выполняется чтение указанного в S1 и S1+1 количества слов или полей, в файле, указанном в S2 (с расширением .IOM, .TXT, CVS), начиная с адреса, указанного в S1+2 и S1+3. Данные записываются в оперативную память, начиная со слова, указанного в D.

Начальный адрес, с которого начинается чтение, указан в S1+2 и S1+3



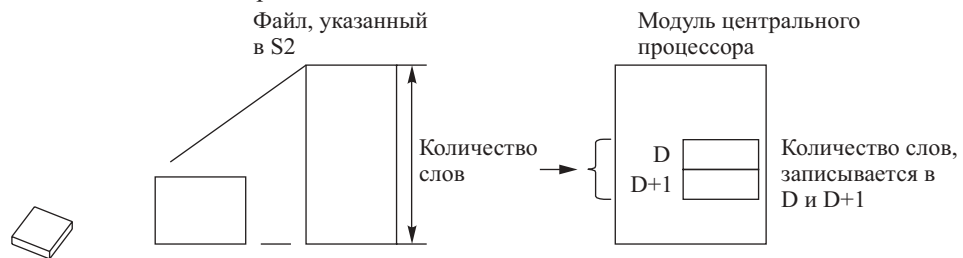
Плата памяти или EM память файлов (указывается первой цифрой слова C)

**Примечание:** Данные запоминаются по порядку истинных адресов внутренней памяти, поэтому если объем прочитанных данных превысит объем области данных, указанной в D, данные будут записываться в следующую область данных, стирая ее содержание. Для дальнейшей информации смотри «Меры предосторожности».

При выполнении команды FREAD(700) количество слов, указанное в S1 и S1+1 записывается в A346 и A347 (количество данных, подлежащих передаче) и это значение уменьшается на единицу после передачи каждого слова или поля. Содержание этих слов можно контролировать для подтверждения передачи ожидаемого количества слов или полей.

**Чтение заданного количества слов данных (третья цифра слова С равна 1)**

Команда FREAD(700) находит количество слов в файле, указанном в S2 (с расширением .IOM) и записывает это восьмизначное шестнадцатеричное число в D и D+1.



Плата памяти или EM память файлов (указывается первой цифрой слова C)

**Спецификации операндов**

Область	C	S1	S2	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6140	CIO 0000...CIO 6143	
Рабочая область	W000...W511	W000...W508	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	H000...H508	H000...H511	
Область вспомогательных битов	A000...A959	A000...A444 A448...A956	A000...A447 A448...A959	A448...A959
Область таймера	T0000...T4095	T0000...T4092	T0000...T4095	
Область счетчика	C0000...C4095	C0000...C4092	C0000...C4095	
Область DM	D00000... D32767	D00000... D32764	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	E00000...E32764	E00000...E32767	
Область EM, содержащая банки	En_00000... En_32767 (n = 0...C)	En_00000... En_32764 (n = 0...C)	En_00000...En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	-	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n = 0...C)		



Область	C	S1	S2	D
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–	*D0000...*D32767 *E0000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	Только указанные значения			
Регистры данных	–			
Индексные регистры	–			
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,-(-)IR0...,-(-)IR15			

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	<p>Переводится в состояние ON, когда указанный в C файл не существует.</p> <p>Переводится в состояние ON, когда установки в слове C выходят за пределы допускаемых значений.</p> <p>Переводится в состояние ON, когда имя файла, указанное в S2 не отвечает требуемым условиям.</p> <p>Переводится в состояние ON, когда флаг работы памяти файлов находится в состоянии ON.</p> <p>Переводится в состояние ON, когда для C не указана константа (не применяется для модели V1).</p> <p>Переводится в состояние ON, когда указанные для S1 данные выходят за пределы допускаемых значений (только для моделей –EV1).</p> <p>Переводится в состояние ON, когда для D указывается недопустимая область.</p> <p>OFF в других случаях.</p>

В следующей ниже таблице показаны относящиеся к команде биты и флаги вспомогательной области.

Наименование	Адрес	Содержание
Тип платы памяти	A34300...A34302	Содержит двоичное число, обозначающее тип платы памяти, если плата установлена.(0: плата не установлена. 4: флэш-память).
Флаг ошибки форматирования платы памяти	A34307	Переводится в состояние ON, когда Плата памяти не форматирована или происходит ошибка форматирования.
Флаг ошибки чтения файла	A34310	Переводится в состояние ON, когда файл не может быть прочитан вследствие повреждения данных или файл содержит другой тип данных.
Флаг безуспешного доступа к файлу	A34311	Переводится в состояние ON, когда данные не могут читаться вследствие отсутствия указанного файла.
Флаг работы памяти файлов	A34313	<p>Переводится в состояние ON в одном из следующих случаев:</p> <p>Модуль центрального процессора передал команду FINS самому себе, используя команду CMND(490).</p> <p>Выполняется команда FREAD(700) или FWRIT(701).</p> <p>Осуществляется перезапись программы с помощью бита управления.</p> <p>Выполняется операция создания резервной копии.</p>
Флаг доступа к файлу	A34314	<p>Переводится в состояние ON при осуществлении доступа к файлу данных.</p> <p>Используйте данный флаг в качестве условия выполнения для предотвращения выполнения команды обработки файла в процессе выполнения другой команды.</p>
Флаг обнаружения Платы памяти	A34315	Переводится в состояние ON при обнаружении Платы памяти.



Наименование	Адрес	Содержание
Начальный банк EM памяти для проведения форматирования	A344	Содержит номер банка области EM, с которого начинается форматирование для использования области в качестве памяти файлов. Содержит FFFF, если форматирование EM области не производилось. Для преобразования области EM в память файлов, параметр EM памяти файлов в начальных установках Программируемого контроллера должен быть установлен в значение, равное 1. Кроме того, необходимо указать начальный банк памяти файлов (от 0 до C). После этого все банки EM памяти, начиная с начального банка до последнего банка, будут отформатированы для использования в качестве памяти файлов.
Флаг ошибки форматирования EM памяти файлов	A34306	Переводится в состояние ON, когда начальный банк EM памяти файлов содержит ошибку форматирования.
Объем данных, подлежащих передаче	A346...A347	Содержание этих слов показывает состояние передачи файлов данных. При выполнении команд FREAD(700) и FWRIT(701) количество слов или полей, подлежащих передаче, записывается в эти слова. Значение уменьшается на единицу после передачи каждого слова или поля. A346 содержит 16 младших битов, а A347 – 16 младших битов двоичного числа длиной 32 бита.

### Меры предосторожности

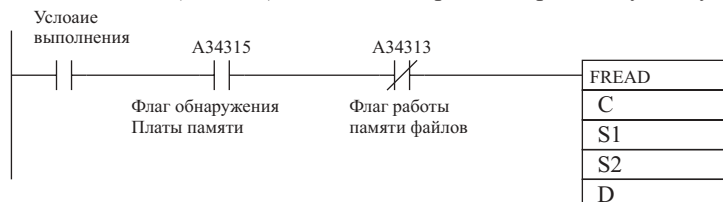
При нормальном выполнении команды, команда FREAD(700) используется только для запуска чтения памяти файлов. Таким образом, время выполнения команд, приведенное в конце данного руководства, является временем, требуемым для запуска чтения, а не для его завершения. В действительности чтение (передача) выполняется посредством осуществления доступа к файлу при периферийном обслуживании. Следовательно, после первого выполнения команды FREAD(700) чтение выполняется длительное время, даже если условие выполнения в последующих циклах находится в состоянии OFF. После завершения передачи флаг работы памяти файлов (A34313) переводится в состояние OFF. Этот флаг может применяться для выполнения эксклюзивного управления командами обработки памяти файлов.

Время, требуемое для завершения передачи данных по команде FREAD(700) зависит от количества передаваемых данных, времени сервисного обслуживания, требуемого для осуществления доступа к файлам, и от других условий. В качестве основного соображения, для цикла длительностью 10 мсек., файла в корневом каталоге и установке времени сервисного обслуживания по умолчанию, время будет равно 0.92 сек. для передачи 1024 слов и 4.64 сек для передачи 9999 слов.

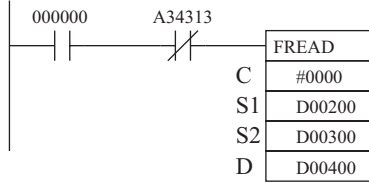
Флаг работы памяти файлов (A34313) после выполнения команды FREAD(700) будет переведен в состояние ON. Если флаг A34313 уже находится в состоянии ON, определяется ошибка и команда выполняться не будет.

Флаг ошибки чтения файла (A34310) переводится в состояние ON и команда не выполняется, если указанный файл содержит неверный тип данных или данные файла повреждены. Для текстовых файлов или файлов CVS, символьный код должен быть шестнадцатеричным числом и разделитель должен вводиться через каждые четыре цифры для данных, состоящих из одного слова, и через каждые восемь цифр для данных, состоящих из двойного слова. Чтение данных будет выполняться до точки определения недействительного символа.

Модуль центрального процессора определяет наличие Платы памяти через несколько секунд после ее установки. Если доступ к Плате памяти должен осуществляться сразу после включения питания, или сразу после установки Платы, для подтверждения обнаружения Платы памяти используйте Флаг обнаружения платы памяти (A34315) в качестве нормально разомкнутого условия ввода, как показано ниже.



**Примеры**



C: #0 0 0 0

Память файлов: Плата памяти.  
Функция: чтение данных

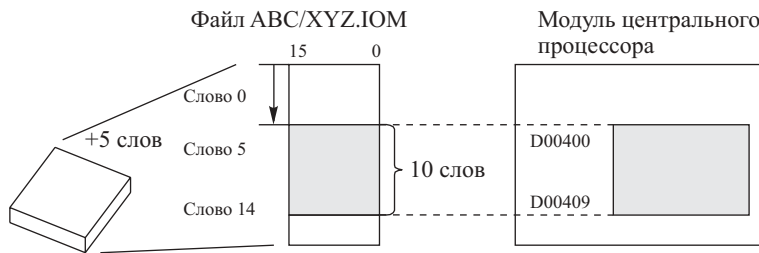
S1: D00200	0	0	0	A
S1+1:D00201	0	0	0	0
S1+2:D00202	0	0	0	5
S1+3:D00203	0	0	0	0

Количество слов, подлежащих чтению: 10 слов

Начальное слово: Начало файла + 5 слов

S2: D00300	5	C	4	1
S2+1:D00301	4	2	4	3
S2+2:D00302	5	C	5	8
S2+3:D00303	5	9	5	A
S2+3:D00304	0	0		Игнорируется

Наименование директории: \ABC  
Имя файла: XYZ



**3-24-2 Команда записи данных в файл WRITE DATA FILE: FWRIT(701)**

**Назначение**

По команде FWRIT(701) осуществляется перезапись или добавление данных в указанный файл данных. Новые данные берутся из указанной области данных Модуля центрального процессора.

**Символ релейно-контактной схемы**

FWRIT(701)	
C	C: Слово управления
D1	D1: Первое слово назначения
D2	D2: Имя файла
S	S: Первое исходное слово

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	FWRIT(701)
	Выполняется один раз для дифференцирования вверх	@ FWRIT(701)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

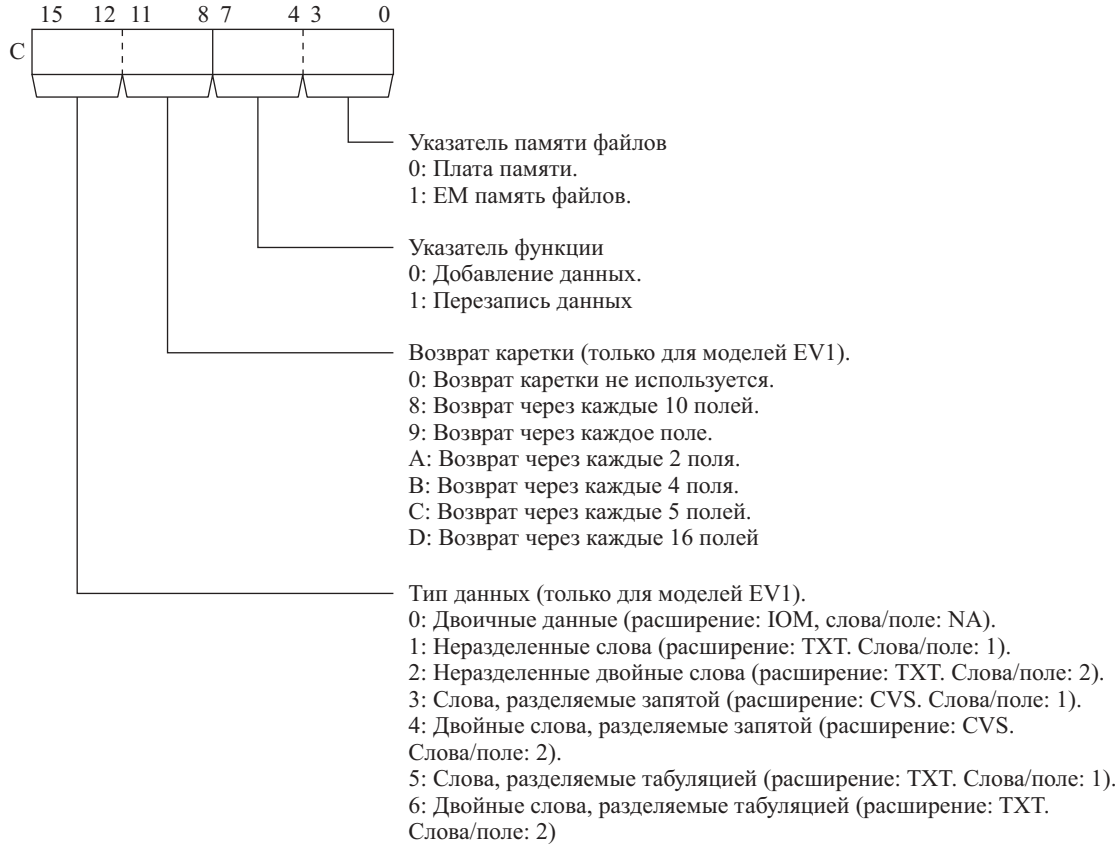
**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды**

**С: Слово управления**

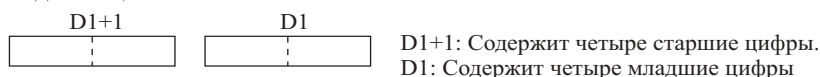
Как показано на следующем ниже рисунке, третья цифра слова управления задает либо перезапись данных, либо добавление данных в файл, а четвертая цифра указывает тип памяти файлов – Плата памяти или ЕМ память файлов.



- Примечание:**
1. Каждое поле содержит одно слово памяти ввода/вывода для данных длиной в одно слово, и два слова памяти ввода/вывода для данных длиной два слова.
  2. При работе с двойными словами первое слово данных читается в адресе памяти, имеющем более высокий номер, т.е. 12345678 будет читаться следующим образом: 1234 в слове D00001, а 5678 – в слове D00000.
  3. При использовании разделителя, указанный разделитель вводится после каждого слова данных, состоящих из одного слова, и после каждых двух слов для данных, состоящих из двойного слова. (Для разделения запятой вводится код запятой, для разделения табуляцией вводится код табуляции.)
  4. Если задается использование неразделенных или двойных слов, данные во всех полях записываются непрерывно без использования разделителя.
  5. При использовании возврата каретки, возврат каретки вводится после каждой последовательности указанного количества слов. Если возврат каретки не используется, данные записываются последовательно.

**D1 и D1+1: Количество записываемых элементов**

Восьмизначное шестнадцатеричное число в словах D+1 и D1+1 указывает количество слов или полей, подлежащих записи.

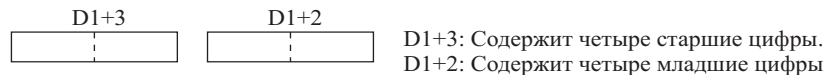


Тип данных	Биты 12 15	Содержание S1 и S1+1
Двоичные данные	0 (шестн.) (двоичные)	Количество слов, подлежащих записи из памяти файлов. 00000000...3FFFFFFF (шестн.)

Тип данных	Биты 12 15	Содержание S1 и S1+1
Слово	1 (шестн.) (неразделенные) 3 (шестн.) (разделенные запятой) 5 (шестн.) (разделенные табуляцией)	Количество полей, подлежащих записи из памяти файлов, т.е. количество слов, подлежащих записи из памяти файлов. 00000000...1FFFFFFF (шестн.)
Двойное слово	2 (шестн.) (неразделенные) 4 (шестн.) (разделенные запятой) 6 (шестн.) (разделенные табуляцией)	Количество полей, подлежащих записи из памяти файлов, т.е. половина количества слов, подлежащих записи из памяти файлов. 00000000...0FFFFFFF (шестн.)

**D1+2 и D1+3: Первое слово назначения**

Восьмизначное шестнадцатеричное число в словах V1+2 и V1+3 указывает первое слово от начала файла, с которого начинается запись.



Тип данных	Биты 12 15	Содержание S1+3 и S1+2
Двоичные данные	0 (шестн.) (двоичные)	Слово, отстоящее от начала файла, с которого начинается запись данных. 00000000...3FFFFFFF (шестн.)
Слово	1 (шестн.) (неразделенные) 3 (шестн.) (разделенные запятой) 5 (шестн.) (разделенные табуляцией)	Поле, отстоящее от начала файла, с которого начинается запись данных, т.е. количество слов, от начала файла. 00000000...1FFFFFFF (шестн.)
Двойное слово	2 (шестн.) (неразделенные) 4 (шестн.) (разделенные запятой) 6 (шестн.) (разделенные табуляцией)	Поле, отстоящее от начала файла, с которого начинается запись данных, т.е. половина количества слов, от начала файла. 00000000...0FFFFFFF (шестн.)

- Примечание:**
- Слова S1+2 и S1+3 используются только для перезаписи данных и только:
    - Для текстовых и CVS данных без возврата каретки (т.е. биты 08...11 слова C устанавливаются в значение, равное 0 (шестн.))
    - Для двоичных данных. Всегда устанавливайте D1+2 и D1+3 в значение 00000000 (шестн.) при записи данных с использованием возврата каретки (т.е. биты 08...11 слова C устанавливайте в значение между 8 и D (шестн.)).
  - Слова D1...D1+3 должны находиться в одной области данных.
  - Если указываемое первое слово превышает количество слов или полей в файле данных, флаг ошибки записи файла (A34308) переводится в состояние ON, и запись данных не производится.

**D2: Имя файла**

D2 является начальным адресом слов, содержащих полное путевое имя и имя файла в кодах ASCII. Используйте коды ASCII от а до z, от А до Z, и от 0 до 9.

Полный путь к каталогу, содержащему файл данных, может содержать до 65 знаков, включая наклонную черту (ASCII 5C). Имя файла может быть длиной до 8 символов, однако имя не должно содержать нулей (ASCII 00), так как ноль используется для обозначения конца символьной строки. Не включайте расширение в имя файла; расширение .IOM, TXT или CVS добавляется автоматически.

D2	F1	F2	Содержат символьную строку, начиная со старшего байта слова D2 Полный путь к каталогу и имя файла может состоять из 74 символов (байтов), включая наклонную черту и ноль, указывающий конец строки
D2+1	F3	F4	
⋮	⋮	⋮	
⋮	⋮	⋮	
D2+38	F73	F74	

- Примечание:**
- Убедитесь в том, что символьная строка, содержащая путь к каталогу и имя файла, не выходит за пределы области данных.
  - Если указываемый файл или каталог не существует, флаг безрезультатного обращения к файлу (A34311) переводится в состояние ON, и запись данных не выполняется.

Записывайте путь и имя файла в кодах ASCII, начиная со старшего байта слова D2, как показано в следующем ниже примере для \ABC\XYZ.IOM. (Расширение .IOM добавляется автоматически.)

D2	"\	"A"	D2	5C	41
D2+1	"B"	"C"	D2+1	42	43
D2+2	"\	"X"	D2+2	5C	58
D2+3	"Y"	"Z"	D2+3	59	5A
D2+4	NUL		D2+4	00	

**S: Первое исходное слово**

S указывает начальный адрес, содержащий данные, которые должны записываться в память файлов. Данные читаются согласно их истинным адресам памяти внутреннего ввода/вывода, поэтому по команде FWRIT(701) производится последовательное чтение исходных данных из следующей области данных, если количество слов, подлежащих чтению, превышает границу области данных, указанной в S.

**Описание**

При нормальном выполнении команды, команда FWRIT(701) используется только для запуска записи памяти файлов. Таким образом, время выполнения команд, приведенное в конце данного руководства, является временем, требуемым для запуска операции, а не для ее завершения. В действительности запись (передача) выполняется посредством осуществления доступа к файлу при периферийном обслуживании. Следовательно, после первого выполнения команды FWRIT(701) запись выполняется длительное время, даже если условие выполнения в последующих циклах находится в состоянии OFF. После завершения передачи флаг работы памяти файлов (A34313) переводится в состояние OFF. Этот флаг может применяться для выполнения эксклюзивного управления командами обработки памяти файлов.

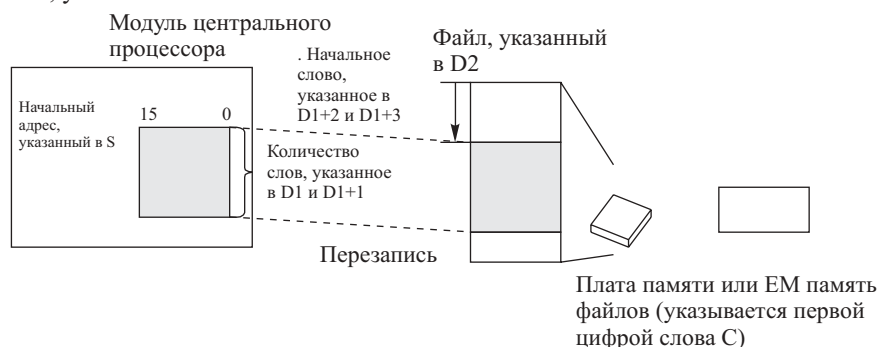
Время, требуемое для завершения передачи данных по команде FWRIT(701) зависит от количества передаваемых данных, времени сервисного обслуживания, требуемого для осуществления доступа к файлам, и от других условий. В качестве основного соображения, для цикла длительностью 10 мсек., файла в корневом каталоге и установке времени сервисного обслуживания по умолчанию, время передачи 1024 слов будет равно 1.97 сек. (для нового файла) или 1.33 сек (для существующего файла). Для передачи 9999 слов это время будет равно 6.64 сек. (для нового файла) или 6.12 сек (для существующего файла).

Исходные данные читаются в оперативной памяти из истинных адресов внутренней памяти, поэтому даже если данные распространяются на две и более областей данных, производится чтение всего блока данных. Например, если первый адрес назначения находится в рабочей области, однако объем данных превышает объем этой области, по команде FWRIT(701) чтение данных продолжается с начала следующей области (в данном случае с начала области таймера). Для ознакомления с расположением областей данных в оперативной памяти обратитесь к карте памяти в Приложении А Руководства по эксплуатации Программируемых контроллеров серии CS1 (W339).

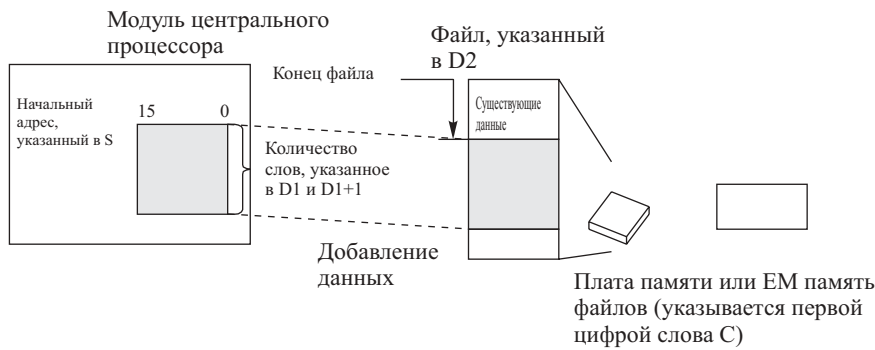
При выполнении команды FWRIT(701) количество слов, указанное в D1 и D1+1 записывается в A346 и A347 (количество данных, подлежащих передаче) и это значение уменьшается на единицу после передачи каждого слова или поля. Содержание этих слов можно контролировать для подтверждения передачи ожидаемого количества слов или полей.

Перезапись данных в существующем файле (третья цифра слова С равна 1)

Команда FWRIT(701) использует область данных, начиная со слова, указанного в S, для перезаписи в памяти файлов данных указанного типа. При этом в файле, указанном в D (с расширением .IOM, .TXT, .CVS), начиная с адреса, указанного в D1+2 и D1+3, производится перезапись заданного количества слов или полей, указанного в D1 и D+1.

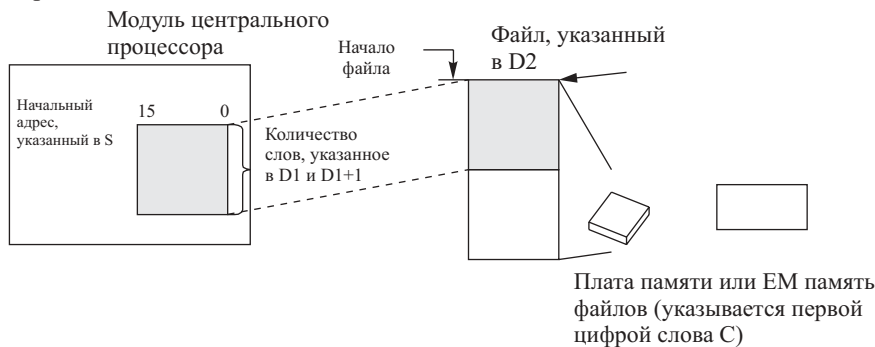
**Добавление данных к существующему файлу (третья цифра слова С равна 0)**

По команде FWRIT(701) производится добавление данных, прочитанных из области данных, начиная со слова S, к данным файла, находящегося в памяти файлов. Тип данных также указывается. При этом количество слов или полей, указанное в D1 и D1+1, добавляется к файлу, указанному в D2 (с расширением .IOM, .TXT, .CVS).



**Создание нового файла, включающего исходные данные**

Если файл, указанный в D2, не существует, по команде FWRIT(701) производится создание нового файла с указанным именем и расширением (.IOM, .TXT, .CVS) и записывает исходные данные указанного типа, начиная с начала файла. В данном случае не существует разницы между указанием добавления данных и перезаписи данных.



**Спецификации операндов**

Область	C	D1	D2	S
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	CIO 0000...CIO 6140	CIO 0000...CIO 6143	
Рабочая область	W000...W511	W000...W508	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	H000...H508	H000...H511	
Область вспомогательных битов	A000...A959	A000...A444 A448...A956	A000...A447 A448...A959	
Область таймера	T0000...T4095	T0000...T4092	T0000...T4095	
Область счетчика	C0000...C4095	C0000...C4092	C0000...C4095	
Область DM	D00000... D32767	D00000... D32764	D00000...D32767	
Область EM, не содержащая банков	E00000...E3276 7	E00000...E3276 4	E00000...E32767	
Область EM, содержащая банки	En_00000... En_32767 (n = 0...C)	En_00000... En_32764 (n = 0...C)	En_00000...En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	—	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	Только указанные значения	—		
Регистры данных	—			
Индексные регистры	—			

Область	C	D1	D2	S
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15			

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	<p>Переводится в состояние ON, когда указанный в C файл не существует.</p> <p>Переводится в состояние ON, когда установки в слове C выходят за пределы допускаемых значений.</p> <p>Переводится в состояние ON, когда имя файла, указанное в D2 не отвечает требуемым условиям.</p> <p>Переводится в состояние ON, когда флаг работы памяти файлов находится в состоянии ON.</p> <p>Переводится в состояние ON, когда для C не указана константа (не применяется для модели V1).</p> <p>Переводится в состояние ON, когда указанные для D1 данные выходят за пределы допускаемых значений (только для моделей –EV1).</p> <p>Переводится в состояние ON, когда для D указывается недопустимая область.</p> <p>OFF в других случаях.</p>

В следующей ниже таблице показаны относящиеся к команде биты и флаги вспомогательной области.

Наименование	Адрес	Содержание
Тип платы памяти	A34300...A34302	Содержит двоичное число, обозначающее тип платы памяти, если плата установлена.(0: плата не установлена. 4: флэш-память).
Флаг ошибки форматирования платы памяти	A34307	Переводится в состояние ON, когда Плата памяти не форматирована или происходит ошибка форматирования.
Флаг ошибки записи файла	A34308	Переводится в состояние ON, когда при записи файла возникает ошибка.
Флаг запрещения записи файла	A34309	Переводится в состояние ON, когда данные не могут быть записаны вследствие защиты его от записи или отсутствует достаточный для записи объем памяти.
Флаг безуспешного доступа к файлу (флаг отсутствия файла)	A34311	Переводится в состояние ON, при записи файла заданная директория отсутствует.
Флаг работы памяти файлов	A34313	<p>Переводится в состояние ON в одном из следующих случаев:</p> <p>Модуль центрального процессора передал команду FINS самому себе, используя команду CMND(490).</p> <p>Выполняется команда FREAD(700) или FWRT(701).</p> <p>Осуществляется перезапись программы с помощью бита управления.</p> <p>Выполняется операция создания резервной копии.</p>
Флаг доступа к файлу	A34314	<p>Переводится в состояние ON при осуществлении доступа к файлу данных.</p> <p>Используйте данный флаг в качестве условия выполнения для предотвращения выполнения команды обработки файла в процессе выполнения другой команды.</p>
Флаг обнаружения Платы памяти	A34315	Переводится в состояние ON при обнаружении Платы памяти.



Наименование	Адрес	Содержание
Начальный банк ЕМ памяти для проведения форматирования	A344	Содержит номер банка области ЕМ, с которого начинается форматирование для использования области в качестве памяти файлов. Содержит FFFF, если форматирование ЕМ области не производилось.  Для преобразования области ЕМ в память файлов, параметр ЕМ памяти файлов в начальных установках Программируемого контроллера должен быть установлен в значение, равное 1. Кроме того, необходимо указать начальный банк памяти файлов (от 0 до С). После этого все банки ЕМ памяти, начиная с начального банка до последнего банка, будут отформатированы для использования в качестве памяти файлов.
Флаг ошибки форматирования ЕМ памяти файлов	A34306	Переводится в состояние ON, когда начальный банк ЕМ памяти файлов содержит ошибку форматирования.
Объем данных, подлежащих передаче	A346...A347	Содержание этих слов показывает состояние передачи файлов данных.  При выполнении команды FWRIT(701) количество слов или полей, подлежащих передаче, записывается в эти слова. Значение уменьшается на единицу после передачи каждого слова или поля.  A346 содержит 16 младших битов, а A347 – 16 младших битов двоичного числа длиной 32 бита.

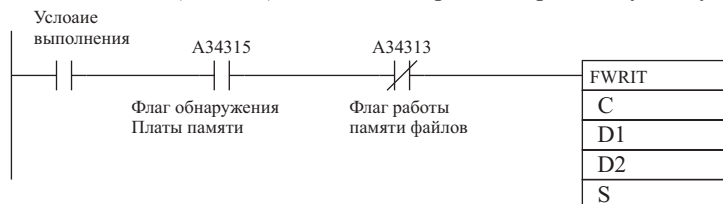
### Меры предосторожности

Флаг работы памяти файлов (A34313) после выполнения команды FWRIT(701) будет переведен в состояние ON. Если флаг A34313 уже находится в состоянии ON, определяется ошибка и команда выполняться не будет.

Флаг запрещения записи файла (A34309) переводится в состояние ON, и команда не выполняется, если данные не могут быть записаны вследствие защиты его от записи или отсутствует достаточный объем памяти.

Флаг ошибки записи файла (A34308) переводится в состояние ON и команда не выполняется, если указанный файл содержит неверный тип данных или данные файла повреждены.

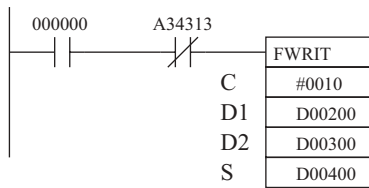
Модуль центрального процессора определяет наличие Платы памяти через несколько секунд после ее установки. Если доступ к Плате памяти должен осуществляться сразу после включения питания, или сразу после установки Платы, для подтверждения обнаружения Платы памяти используйте Флаг обнаружения платы памяти (A34315) в качестве нормально разомкнутого условия ввода, как показано ниже.



### Примеры

В следующем ниже примере, когда CIO 000000 переводится в состояние ON, по команде FWRIT(701) производится чтение 10 слов данных в D00400...D00409 и эти данные используются для перезаписи 10 слов в файле \ABC\XYZ.IOM, начиная с начала файла + 5 слов.





C: #0 0 1 0

Память файлов: Плата памяти.  
Функция: Перезапись данных

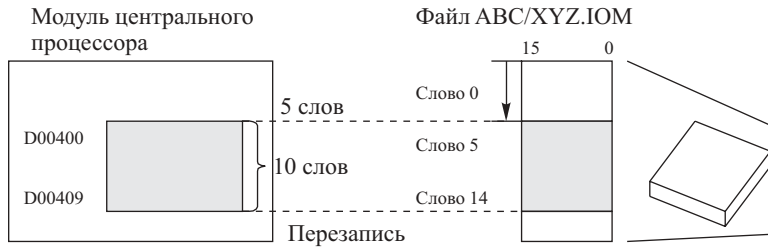
D1: D00200	0	0	0	A
D1+1:D00201	0	0	0	0
D1+2:D00202	0	0	0	5
D1+3:D00203	0	0	0	0

Количество слов, подлежащих записи: 10 слов

Начальное слово: Начало файла + 5 слов

D2: D00300	5	C	4	1
D2+1:D00301	4	2	4	3
D2+2:D00302	5	C	5	8
D2+3:D00303	5	9	5	A
D2+3:D00304	0	0		Игнорируется

Наименование директории: \ABC  
Имя файла: XYZ



## 3-25 Команды дисплея: DISPLAY MESSAGE: MSG(046)

**Назначение**

По команде MSG(046) осуществляется чтение указанных шестнадцати слов в расширенном коде ASCII и вывод сообщения в периферийное устройство, например Пульт программирования.

**Символ релейно-контактной схемы**

MSG(046)	
N	N: Номер сообщения
M	M: Первое слово сообщения

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	MSG(046)
	Выполняется один раз для дифференцирования вверх	@ MSG(046)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****N: Номер сообщения**

Номер сообщения должен находиться в пределах от 0000 до 0007 (шестн.) (от 0 до 7 в десятичном коде).

**M: Первое слово сообщения**

При выводе сообщения M указывает адрес первого из слов, содержащих сообщение в ASCII коде. При стирании сообщения M может быть любой постоянной в шестнадцатеричном коде (от 0000 до FFFF).

**Спецификации операндов**

Область	N	M
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A000...A959	
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n=0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n=0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)	
Константы	#0000...#0007 (двоичные) или &0...&7	#0000...#0007 (двоичные)
Регистры данных	DR0...DR15	—
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(++) ,(-)IR0...,-(-)IR15	

**Описание**

Когда условие выполнения находится в состоянии ON, по команде MSG(046) производится регистрация 16-ти слов данных в ASCII коде (до 32 символов, включая ноль) от M до M+16. Номер сообщения указыва-

ется в N. После того, как сообщение зарегистрировано, можно подключать Пульт программирования, и после появления сообщений об ошибках их можно читать на дисплее.

После регистрации сообщения содержание дисплея может изменяться посредством перезаписи сообщения в области хранения сообщений.

Для удаления зарегистрированного сообщения выполните команду MSG(046) изменив содержание слова S на значение равное номеру сообщения, которое необходимо удалить, и установив в N константу (0000...FFFF).

Сообщение, регистрируемое в процессе выполнения команды, будет сохранено, даже если выполнение программы прерывается. Тем не менее, все сообщения удаляются при повторном запуске программы.

**Примечание:** Для ознакомления с расширенной таблицей кода ASCII обратитесь к **Приложению А Руководства по эксплуатации Пульта программирования серии CS1.**

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, если содержание слова C выходит за пределы допустимых значений от 0000 до 0007 (шестн.). OFF в других случаях.

#### Меры предосторожности

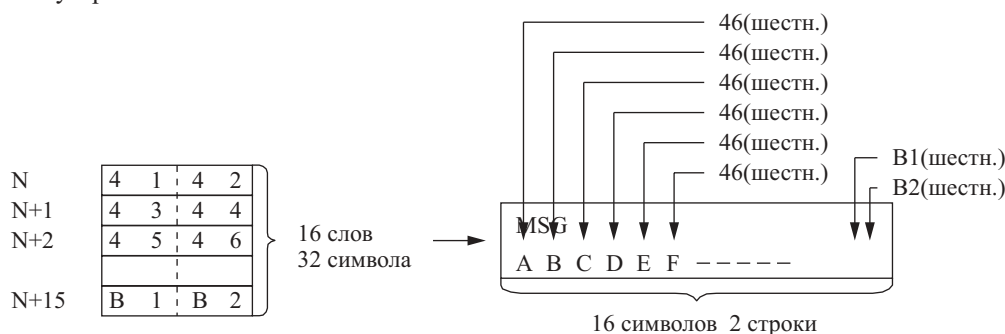
Зарегистрированные сообщения обновляются при каждом выполнении команды MSG(064).

Все символы сообщения после нуля (00) на дисплее Пульта программирования преобразуются в пробелы. Символы, сохраняемые в старших байтах, выводятся на дисплей раньше символов, сохраняемых в младших байтах.

#### Примеры

На следующем ниже рисунке показан пример преобразования 16-ти слов шестнадцатеричных данных в сообщение, выводимое на дисплей Пульта программирования.

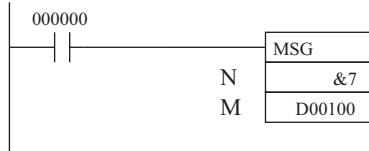
В следующем(п = 0...С)ниже примере, когда CIO 000000 переводится в состояние ON, 16 слов данных в D00100...D00115 читаются в виде 32-х символов ASCII кода для сообщения 7, и выводятся в Периферийное устройство.



#### Таблица кодов ASCII

Четыре правых бита	Четыре левых бита										A	B	C	D	E	F
	0	1	2	3	4	5	6	7	8	9						
0			Sp	0	@	P	`	p			Японские иероглифы					
1			!	1	A	Q	a	q								
2			"	2	B	R	b	r								
3			#	3	C	S	c	s								
4			\$	4	D	T	d	t								
5			%	5	E	U	e	u								
6			&	6	F	V	f	v								
7			'	7	G	W	g	w								
8			(	8	H	X	h	x								
9			)	9	I	Y	i	y								
A			*	:	J	Z	j	z								
B			+	;	K	[	k	{								
C			,	<	L		l									
D			-	=	M	]	m	}								

Четыре правых бита	Четыре левых бита															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
E			.	>	N	^	n	~								
F			/	?	O	_	o									



### 3-26 Команды управления часами

Настоящий раздел приводит описание команд, используемых для управления системными часами.

Команда	Мнемоническое обозначение	Функциональный код	Страница
CALENDAR ADD	CADD	730	638
CALENDAR SUBSTRACT	CSUB	731	641
HOURS TO SECONDS	SEC	065	638
SECONDS TO HOURS	HMS	066	646
CLOCK ADJUSTMENT	DATE	735	648

#### 3-26-1 Команда установки календаря CALENDAR ADD: CADD(730)

##### Назначение

По команде CADD(730) осуществляется суммирование данных времени и данных календаря, находящихся в указанных словах.

##### Символ релейно-контактной схемы

CADD(730)	
C	C: Первое слово календаря
T	N: Первое слово времени
R	R: Первое слово результата

##### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	CADD(730)
	Выполняется один раз для дифференцирования вверх	@ CADD(730)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

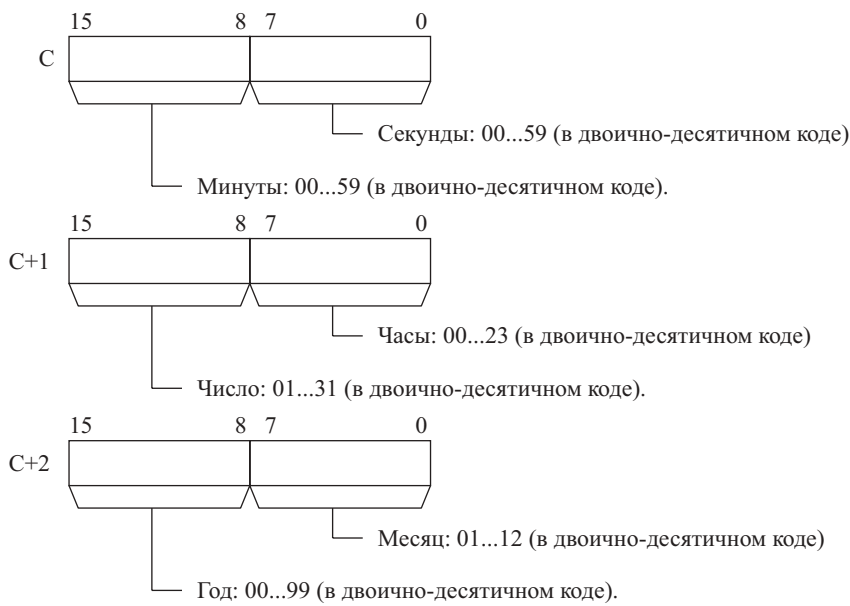
##### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

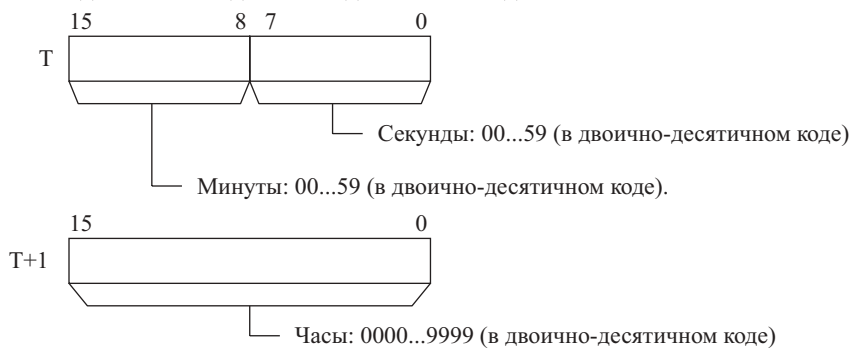
##### Операнды

##### От C до C+2: Данные календаря

Устанавливайте данные календаря в словах C...C+2, как показано на следующем ниже рисунке. Слова C...C+2 должны находиться в одной области данных.

**Т и Т+1: Данные времени**

Устанавливайте данные о времени в словах Т и Т+1, как показано на следующем ниже рисунке. Слова Т и Т+1 должны находиться в одной области данных.

**От R до R+2: Результирующие данные**

Слова R...R+2 содержат результат суммирования. Слова R...R+2 должны находиться в одной области данных.



### Спецификации операндов

Область	C	T	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6141	CIO 0000...CIO 6142	CIO 0000...CIO 6141
Рабочая область	W000...W509	W000...W510	W000...W509
Область удержания бита (Holding bit area)	H000...H509	H000...H510	H000...H509
Область вспомогательных битов	A000...A957	A000...A958	A448...A957
Область таймера	T0000...T4093	T0000...T4094	T0000...T4093
Область счетчика	C0000...C4093	C0000...C4094	C0000...C4093
Область DM	D00000...D32765	D00000...D32766	D00000...D32765
Область EM, не содержащая банков	E00000...E32765	E00000...E32766	E00000...E32765
Область EM, содержащая банки	En_00000...En_32765 (n = 0...C)	En_00000...En_32766 (n = 0...C)	En_00000...En_32765 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	–	Только указанные значения	–
Регистры данных	–		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,-( - )IR15		

### Описание

По команде CADD(730) осуществляется суммирование данных календаря (слова C...C+2) и данных времени (слова T T T+1). Результат выводится в слова R...R+2.

C	15	8	7	0
	Минуты	Секунды		
C+1	Число		Час	
C+2	Год		Месяц	

+

T	15	8	7	0
	Минуты	Секунды		
T+1	Часы			

↓

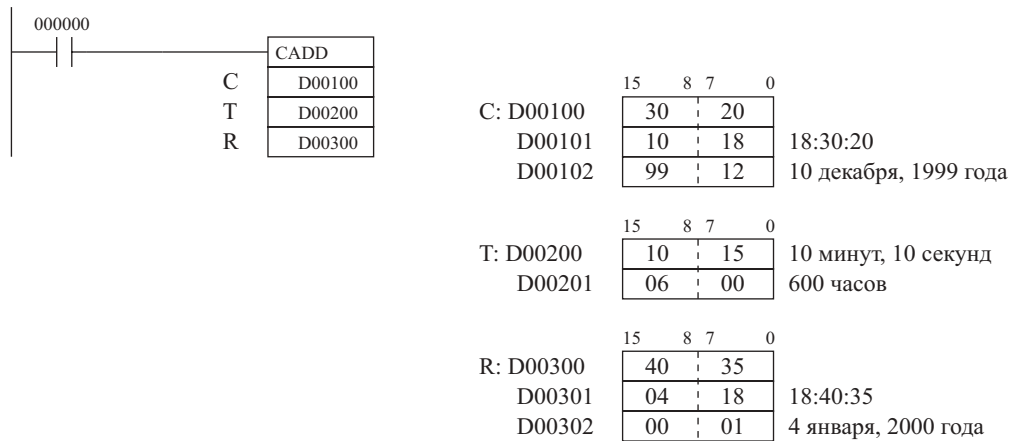
R	15	8	7	0
	Минуты	Секунды		
R+1	Число		Час	
R+2	Год		Месяц	

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда данные календаря в словах C...C+2 выходят за пределы допускаемых значений. Переводится в состояние ON, когда данные времени в словах T и T+1 выходят за пределы допускаемых значений. OFF в других случаях.

### Примеры

В следующем ниже примере, когда CIO 000000 переводится в состояние ON, данные календаря в словах D00100...D00102 (год, месяц, число, часы, минуты, секунды) суммируются с данными времени в словах D00200...D00201 (часы, минуты, секунды). Результат выводится в слова D00300...D00302.

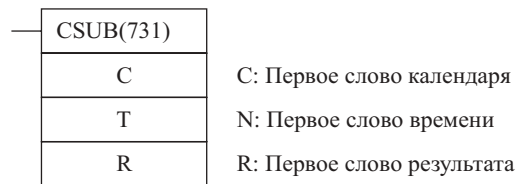


### 3-26-2 Команда установки календаря CALENDAR SUBSTRACT: CSUB(731)

#### Назначение

По команде CSUB(731) осуществляется вычитание данных времени из данных календаря, находящихся в указанных словах.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	CSUB(731)
	Выполняется один раз для дифференцирования вверх	@ CSUB(731)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

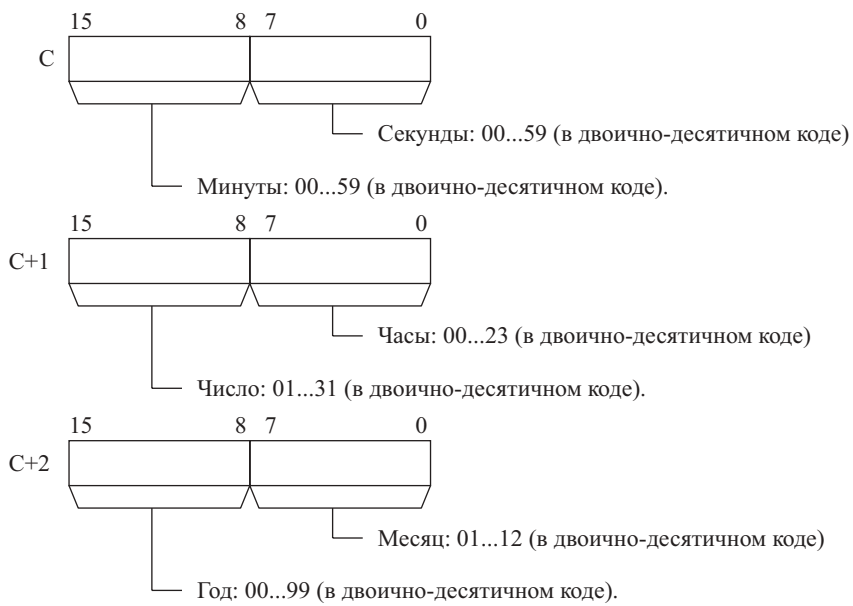
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

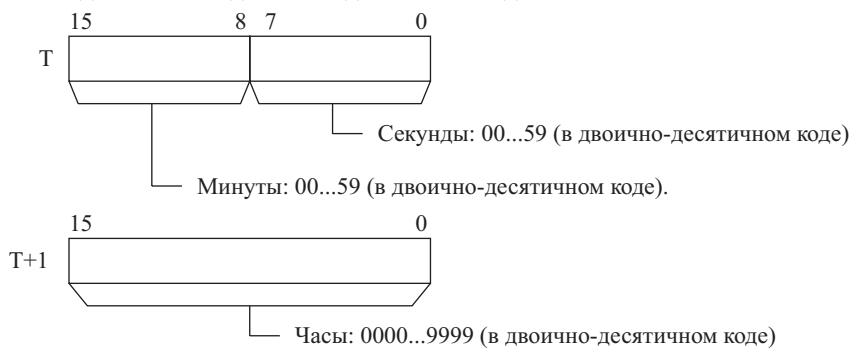
##### От C до C+2: Данные календаря

Устанавливайте данные календаря в словах C...C+2, как показано на следующем ниже рисунке. Слова C...C+2 должны находиться в одной области данных.



**Т и Т+1: Данные времени**

Устанавливайте данные о времени в словах Т и Т+1, как показано на следующем ниже рисунке. Слова Т и Т+1 должны находиться в одной области данных.

**От R до R+2: Результирующие данные**

Слова R...R+2 содержат результат вычитания. Слова R...R+2 должны находиться в одной области данных.



### Спецификации операндов

Область	C	T	R
Область ввода/вывода (область CIO)	CIO 0000...CIO 6141	CIO 0000...CIO 6142	CIO 0000...CIO 6141
Рабочая область	W000...W509	W000...W510	W000...W509
Область удержания бита (Holding bit area)	H000...H509	H000...H510	H000...H509
Область вспомогательных битов	A000...A957	A000...A958	A448...A957
Область таймера	T0000...T4093	T0000...T4094	T0000...T4093
Область счетчика	C0000...C4093	C0000...C4094	C0000...C4093
Область DM	D00000...D32765	D00000...D32766	D00000...D32765
Область EM, не содержащая банков	E00000...E32765	E00000...E32766	E00000...E32765
Область EM, содержащая банки	En_00000...En_32765 (n = 0...C)	En_00000...En_32766 (n = 0...C)	En_00000...En_32765 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n = 0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)		
Константы	–	Только указанные значения	–
Регистры данных	–		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,-(--)IR15		

### Описание

По команде CADD(730) осуществляется вычитание данных времени (слова T т T+1) из данных календаря (слова C...C+2). Результат выводится в слова R...R+2.

	15	8	7	0
C	Минуты		Секунды	
C+1	Число		Час	
C+2	Год		Месяц	

	15	8	7	0
T	Минуты		Секунды	
T+1	Часы			

↓

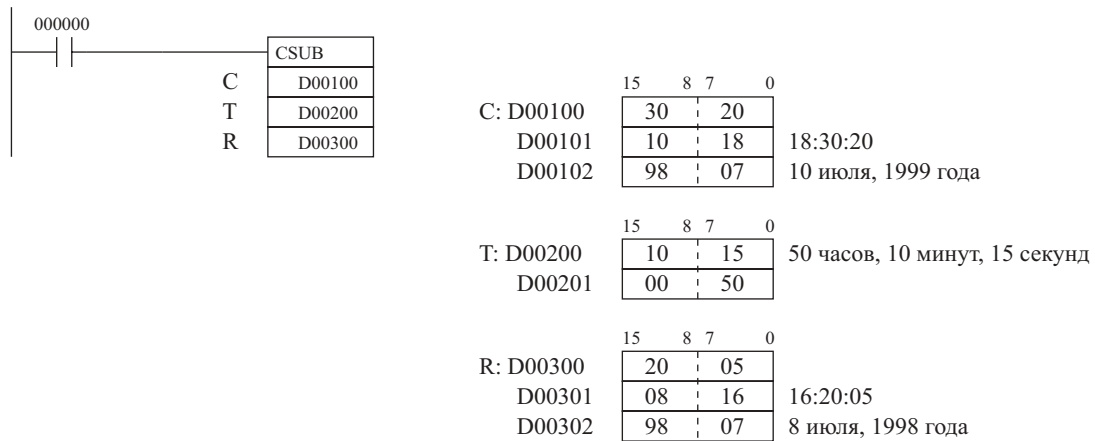
	15	8	7	0
R	Минуты		Секунды	
R+1	Число		Час	
R+2	Год		Месяц	

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда данные календаря в словах C...C+2 выходят за пределы допускаемых значений. Переводится в состояние ON, когда данные времени в словах T и T+1 выходят за пределы допускаемых значений. OFF в других случаях.

### Примеры

В следующем ниже примере, когда CIO 000000 переводится в состояние ON, данные времени в словах D00200...D00201 (часы, минуты, секунды) вычитаются из данных календаря в словах D00100...D00102 (год, месяц, число, часы, минуты, секунды). Результат выводится в слова D00300...D00302.

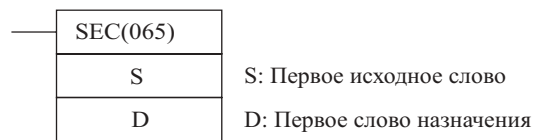


3-26-3 Команда преобразования данных времени из часов в секунды **HOURS TO SECONDS: SEC(065)**

**Назначение**

По команде SEC(065) осуществляется преобразование времени, выраженного в формате часы/минуты/секунды, в эквивалентное время, выраженное в только в секундах.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	SEC(065)
	Выполняется один раз для дифференцирования вверх	@ SEC(065)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

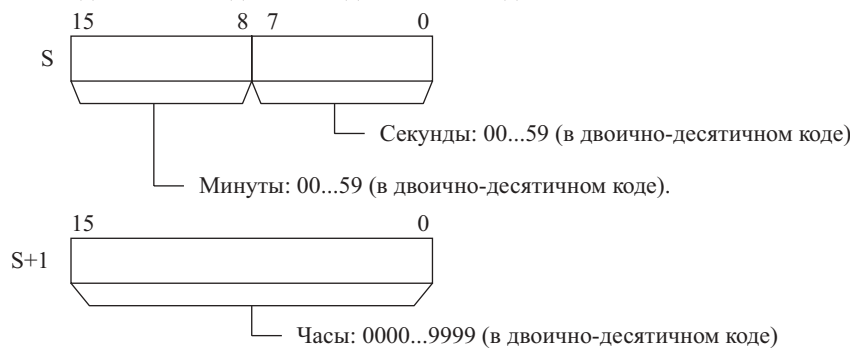
**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды**

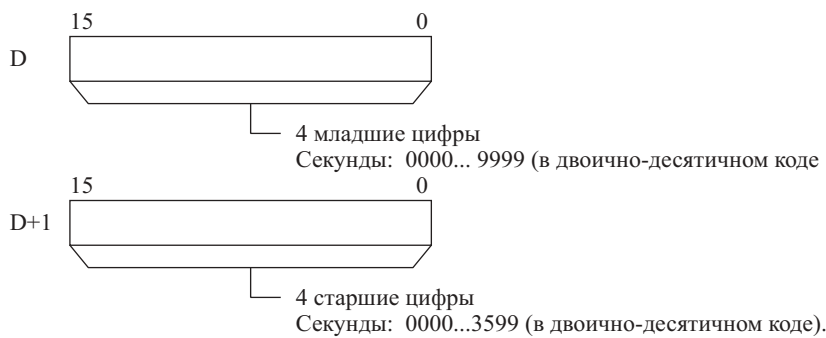
**S и S+1: Исходные данные**

Устанавливайте исходные данные в словах S и S+1, как показано на следующем ниже рисунке. Слова S и S+1 должны находиться в одной области данных.



**D и D+1: Результирующие данные**

Слова D и D+1 содержат результирующие данные, выраженные в секундах. Слова D и D+1 должны находиться в одной области данных.

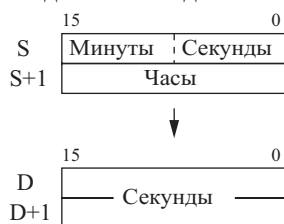


### Спецификации операндов

Область	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W509	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	Только указанные значения	–
Регистры данных	–	
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15	

### Описание

По команде SEC(065) осуществляется преобразование восьмизначных двоично-десятичных данных времени, содержащихся в словах S и S+1 и выраженных в часах/минутах/секундах, в восьмизначные двоично-десятичные данные времени, выраженных в секундах. Результат выводится в слова D и D+1.



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда данные минут в слове S (биты от 08 до 15) выражены не в двоично-десятичном коде или выходят за пределы допускаемых значений от 00 до 59. Переводится в состояние ON, когда данные секунд в слове S (биты от 00 до 07) выражены не в двоично-десятичном коде или выходят за пределы допускаемых значений от 00 до 59. OFF в других случаях.

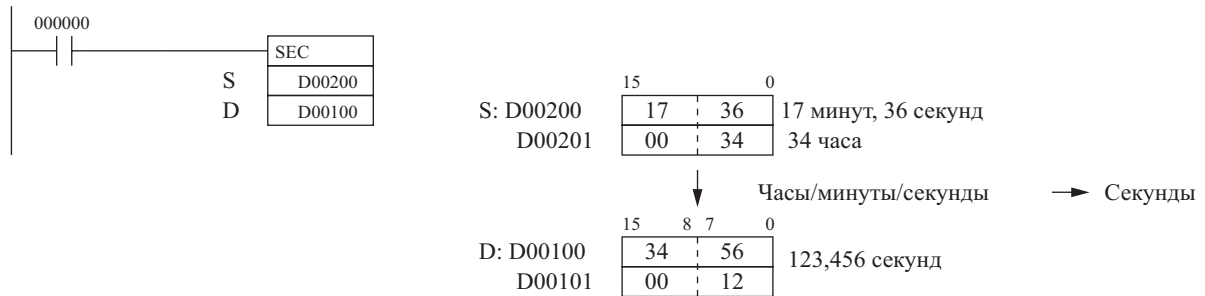
Наименование	Обозначение	Функционирование
Флаг равенства	=	Переводится в состояние ON, когда после выполнения операции содержание D равно 0000. OFF в других случаях.

**Меры предосторожности**

Максимальными значениями исходных данных являются: 9999 часов, 59 минут, 59 секунд (35999999 секунд).

**Примеры**

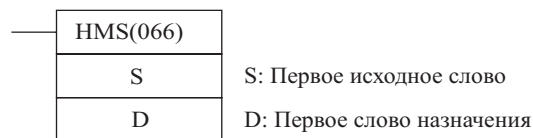
В следующем ниже примере, когда CIO 000000 переводится в состояние ON, данные в словах D00200 и D00201 (34 часа, 17 минут, 36 секунд) преобразуются в данные, выраженные в секундах. Результат выводится в слова D00100 и D00101.



### 3-26-4 Команда преобразования данных времени из секунд в часы SECONDS TO HOURS: HMS(066)

**Назначение**

По команде HMS(066) осуществляется преобразование времени, выраженного в секундах, в эквивалентное время, выраженное в формате часы/минуты/секунды.

**Символ релейно-контактной схемы****Модификации**

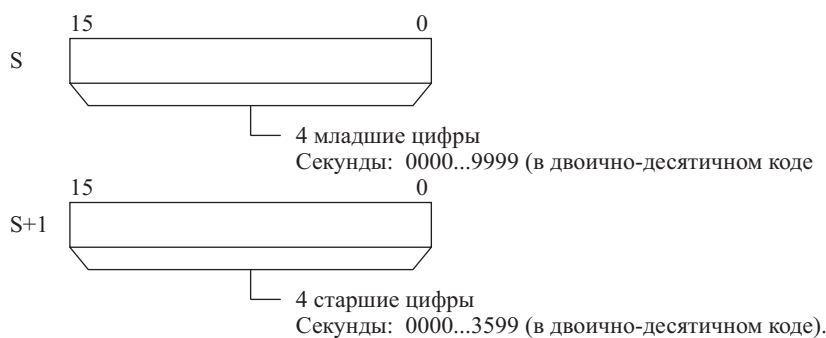
Модификации	Выполняется в каждом цикле для состояния ON	HMS(066)
	Выполняется один раз для дифференцирования вверх	@ HMS(066)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

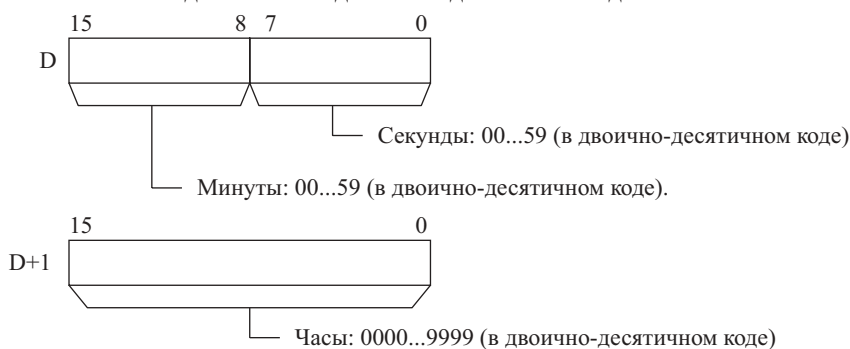
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****S и S+1: Исходные данные**

Устанавливайте исходные данные в словах S и S+1, выраженные в секундах, как показано на следующем ниже рисунке. Слова S и S+1 должны находиться в одной области данных.

**D и D+1: Результирующие данные**

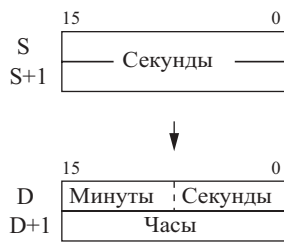
Слова D и D+1 содержат результирующие данные, выраженные в формате часы/минуты/секунды. Слова D и D+1 должны находиться в одной области данных.

**Спецификации операндов**

Область	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6142	
Рабочая область	W000...W509	
Область удержания бита (Holding bit area)	H000...H510	
Область вспомогательных битов	A000...A958	A448...A958
Область таймера	T0000...T4094	
Область счетчика	C0000...C4094	
Область DM	D00000...D32766	
Область EM, не содержащая банков	E00000...E32766	
Область EM, содержащая банки	En_00000...En_32766 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n = 0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n = 0...C)	
Константы	00000000...35999999 (двоично-десятичные)	—
Регистры данных	—	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,(-)IR15	

**Описание**

По команде HMS(066) осуществляется преобразование восьмизначных двоично-десятичных данных времени, содержащихся в словах S и S+1 и выраженных в секундах, в восьмизначные двоично-десятичные данные времени, выраженных в часах/минутах/секундах. Результат выводится в слова D и D+1.



**Флаги**

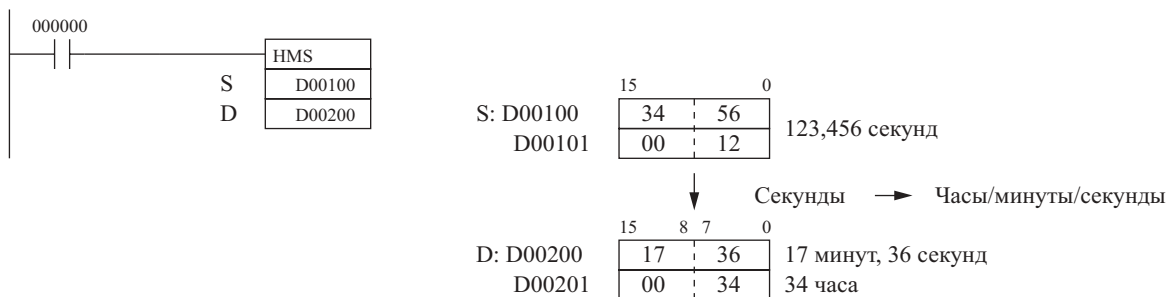
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда данные в словах S и S+1 выражены не в двоично-десятичном коде или выходят за пределы допускаемых значений от 0 до 35999999. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда после выполнения операции содержание D равно 0000. OFF в других случаях.

**Меры предосторожности**

Максимальным значением исходных данных является: 359999999 секунд (9999 часов, 59 минут, 59 секунд).

**Примеры**

В следующем ниже примере, когда СЮ 000000 переводится в состояние ON, данные, выраженные в секундах и находящиеся в словах D00100 и D00101 (123456 секунд) преобразуются в данные, выраженные в формате часы/минуты/секунды. Результат выводится в слова D00200 и D00201.



3-26-5 Команда установки даты CLOCK ADJUSTMENT: DATE(735)

**Назначение**

По команде DATE(735) производится изменение установок внутренних часов на установки, находящиеся в указанных словах.

**Символ релейно-контактной схемы**



**Модификации**

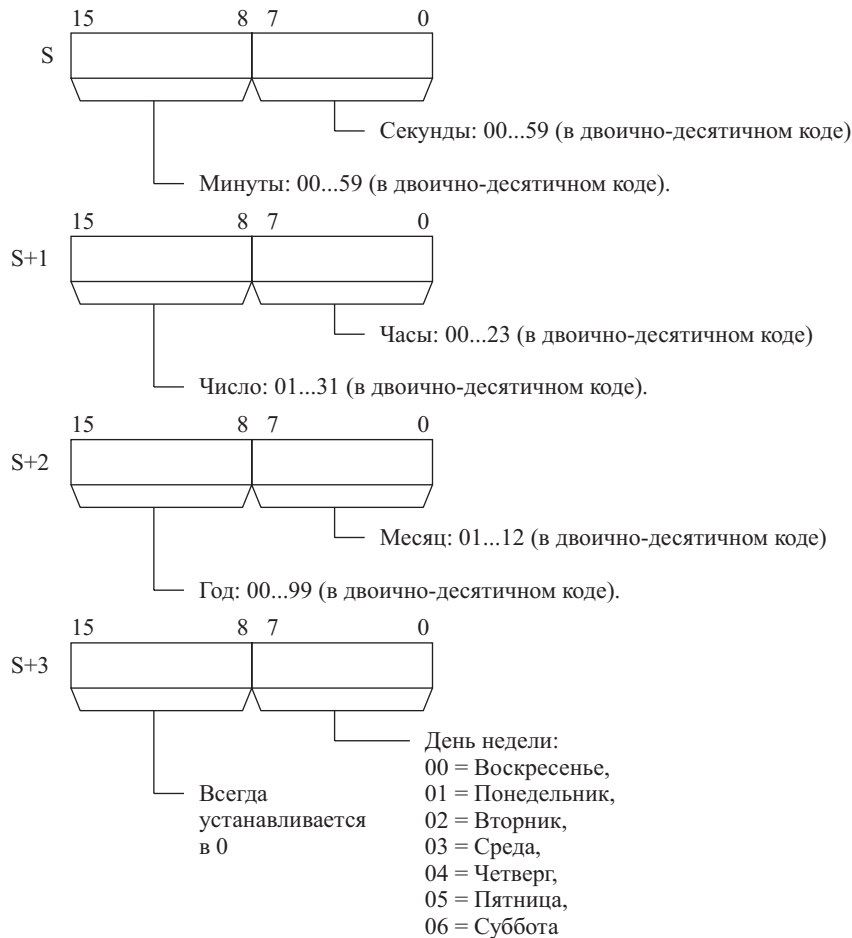
Модификации	Выполняется в каждом цикле для состояния ON	DATE(735)
	Выполняется один раз для дифференцирования вверх	@ DATE(735)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****От S до S+3: Новые установки часов**

Устанавливайте новые данные часов в словах S...S+3, как показано на следующем ниже рисунке. Слова S...S+3 должны находиться в одной области данных.



В следующей ниже таблице показана структура области календаря/часов.

Адреса	Содержание
A35100...A35107	Секунды (00...59, в двоично-десятичном коде)
A35108...A35115	Минуты (00...59, в двоично-десятичном коде)
A35200...A35207	Часы (00...23, в двоично-десятичном коде)
A35208...A35215	Число месяца (01...31, в двоично-десятичном коде)
A35300...A35307	Месяц (01...12, в двоично-десятичном коде)
A35308...A35315	Год (00...99, в двоично-десятичном коде)
A35400...A35407	День недели (00...06 = Воскресенье...Суббота в шестнадцатеричном коде)
A35408...A35415	Всегда устанавливайте в значение, равное 00.

**Спецификации операндов**

Область	C
Область ввода/вывода (область CIO)	CIO 0000...CIO 6140
Рабочая область	W000...W508
Область удержания бита (Holding bit area)	H000...H508
Область вспомогательных битов	A000...A956
Область таймера	T0000...T4092
Область счетчика	C0000...C4092
Область DM	D00000...D32764
Область EM, не содержащая банков	E00000...E32764



Область	C
Область EM, содержащая банки	En_00000...En_32764 (n = 0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767
Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...IR15(++) ,(- -)IR0...(- -)IR15

### Описание

По команде DATE(735) производится изменение установок внутренних часов на установки, находящиеся в четырех исходных словах. Новые данные часов немедленно отражаются в области календаря/часов (A351...A354).

Модуль центрального процессора



### Флаги

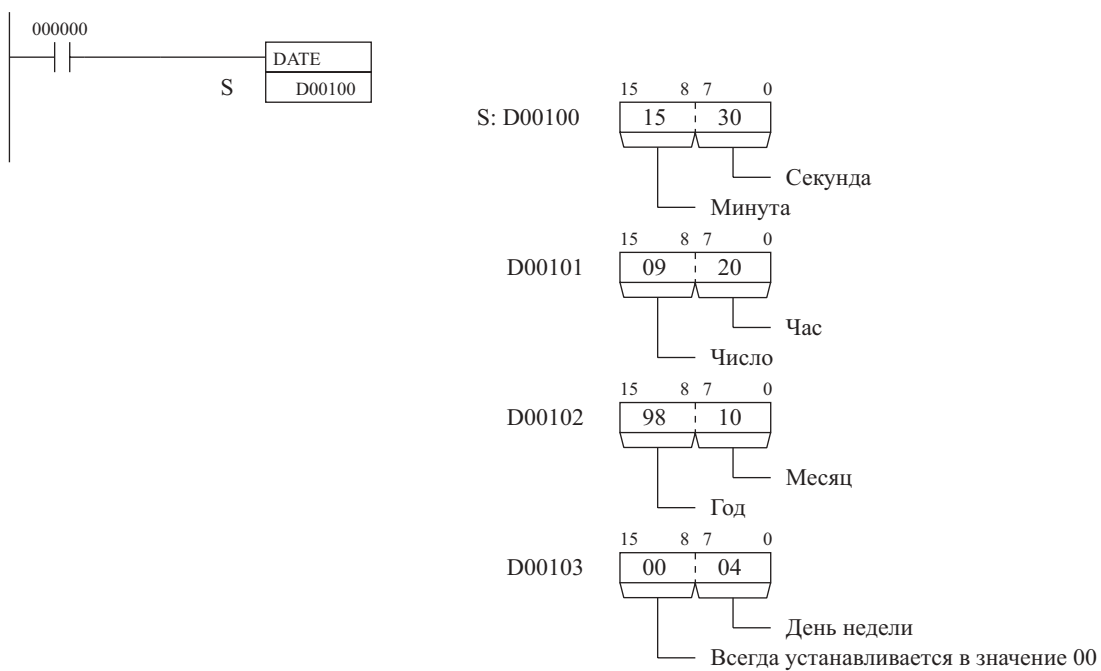
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда новые данные часов в словах S и S+1 выходят за пределы допустимых значений. OFF в других случаях.

### Меры предосторожности

В случае, когда во внутренние часы вводится несуществующая дата (например, 31 Ноября), определяется ошибка.

### Примеры

В следующем ниже примере, когда CIO 000000 переводится в состояние ON, внутренние часы устанавливаются в состояние: 20:15:30, Вторник, 9 Октября, 1998.



## 3-27 Команды поиска и устранения неисправностей

## 3-27-1 Команда Trace Memory Sampling: TRSM(045)

**Назначение**

При выполнении команды TRSM(045) состояние выбранного бита или слова читается и запоминается в памяти отслеживания данных. Эта команда может использоваться любое количество раз в любом месте программы.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	TRSM(045)
	Выполняется один раз для дифференцирования вверх	@ TRSM(045)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков программы	Области шагов программы	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Описание**

Перед выполнением команды TRSM(045), бит или слово, которые подлежат отслеживанию, должны быть указаны с помощью Периферийного устройства. Каждый раз при выполнении команды TRSM(045) текущее значение указанного бита или слова читается и по порядку записывается в память отслеживания данных.

Отслеживание данных прекращается после заполнения памяти. Содержание памяти отслеживания данных при необходимости можно просмотреть с помощью Периферийного устройства.

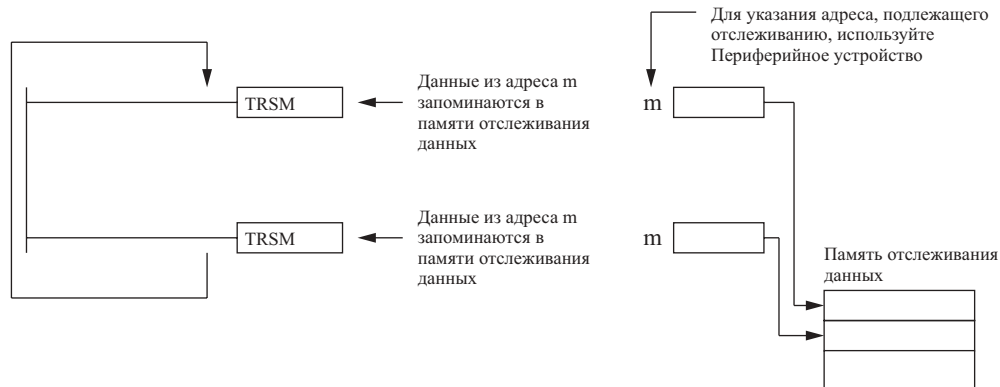


Команда TRSM(045) только обозначает, когда будет производиться выборка данных. Все прочие установки и операции выполнения отслеживания данных устанавливаются с помощью Периферийного устройства. Другими путями отслеживания данных являются выборка данных в конце каждого цикла и выборка данных через указанный интервал времени (независимо от длительности цикла).

Команда TRSM(045) не требует условия применения и всегда выполняется, как если бы ее условие выполнения всегда находилось в состоянии ON. Команда TRSM(045) соединяется непосредственно с левой линией шины.

Используйте команду TRSM(045) для выборки значения указанного бита или слова в точке программы, где условие выполнения команды находится в состоянии ON. Если условие выполнения команды переводится в состояние ON в каждом цикле, значение указанного бита или слова будет запоминаться в памяти отслеживания данных в каждом цикле.

Существует возможность объединения двух и более команд TRSM(045) в одной программе. В этом случае значение одного и того же указанного бита или слова будет запоминаться в памяти отслеживания данных при выполнении каждой из команд TRSM(045).



**Примечание:** Для детального ознакомления с процедурой отслеживания данных обратитесь к Руководству по эксплуатации Периферийного устройства.

Операции, выполняемые с помощью Периферийного Устройства при отслеживании данных, сведены в следующий перечень.

- 1,2,3...**
- С помощью Периферийного устройства задайте следующие ниже параметры.
    - Задайте адрес бита или слова, подлежащего отслеживанию.
    - Задайте условие запуска. Одно из следующих условий может управлять процессом подтверждения записанных в память данных:
      - Бит начала отслеживания данных переводится из состояния OFF в состояние ON.
      - Указанный бит переводится из состояния OFF в состояние ON.
      - Значение указанного слова соответствует заданному значению.
    - Задайте интервал выборки данных в значение «TRSM» для выборки данных при выполнении команды TRSM.
    - Задайте величину задержки.
  - Когда с помощью Периферийного устройства Бит запуска выборки данных переводится из состояния OFF в состояние ON, выборка указанных данных осуществляется при каждом выполнении команды TRSM(045) и выбранные данные запоминаются в Памяти отслеживания данных. В это же время флаг отслеживания данных (Trace Busy Flag A50813) переводится в состояние ON.
  - Когда возникает условие запуска (Бит запуска отслеживания данных в состоянии ON, указанный бит в состоянии ON, или значение указанного слова соответствует заданному значению), отслеживаемые данные принимаются в качестве действительных данных, начиная со следующей выборки плюс или минус некоторое количество выборок, определяемое заданной задержкой. В это же время флаг монитора запуска отслеживания данных (Trace Trigger Monitor Flag A50811) переводится в состояние ON.
  - Отслеживание данных прекращается, когда команда TRSM(045) выполняется достаточное количество раз для заполнения памяти отслеживания данных. После прекращения отслеживания данных флаг завершения отслеживания данных (Trace Completed Flag A 50812) переводится в состояние ON, а флаг отслеживания данных (Trace Busy Flag A50813) переводится в состояние OFF.
- С помощью Периферийного устройства прочитайте содержание памяти отслеживания данных.

В следующей ниже таблице приведены биты и флаги вспомогательной области, относящиеся к данной команде. Только биты A50814 и A50815 могут управляться пользователем, а бит A50815 не должен переводиться в состояние ON из программы, т.е. может изменять состояние только из Периферийного устройства.

Наименование	Адрес	Функционирование
Флаг монитора запуска отслеживания данных Trace Trigger Monitor Flag	A50811	Этот флаг переводится в состояние ON, когда изменением состояния бита запуска отслеживания данных обеспечиваются условия запуска. Флаг переводится в состояние OFF, когда выборка данных возобновляется при следующем отслеживании данных (с помощью бита запуска выборки данных).
Флаг завершения отслеживания данных Trace Completed Flag	A50812	Флаг переводится в состояние ON, когда при осуществлении выборки и запоминания данных полностью заполняется. Флаг переводится в состояние OFF, когда бит запуска выборки данных переходит из состояния OFF в состояние ON.
Флаг отслеживания данных Trace Busy Flag	A50813	Флаг переводится в состояние ON, когда бит запуска выборки данных переходит из состояния OFF в состояние ON. Флаг переводится в состояние OFF после завершения отслеживания данных.
Бит запуска отслеживания данных Trace Start Bit	A50814	Условия запуска отслеживания данных обеспечиваются, когда данный бит переводится из состояния OFF в состояние ON. Данные выборки записываются после истечения времени задержки (положительная задержка), или подтверждения указанного количества существующих выборок данных (отрицательная задержка).
Бит запуска выборки данных Sampling Start bit	A50815	Когда данный бит с помощью Периферийного устройства переводится из состояния OFF в состояние ON, осуществляется запись выборок данных в память отслеживания данных. При этом используется один из методов выборки данных: Периодическая выборка данных (с интервалом от 10 до 2550 мсек.). Выборка данных при выполнении команды TRSM(045). Выборка данных в конце каждого периода. Изменение состояния данного бита должно осуществляться из Периферийного устройства.

**Меры предосторожности**

Команда TRSM(045) выполняется как команда NOP(000), когда отслеживание данных не производится, или интервал выборки данных, установленный в параметрах с помощью Периферийного устройства, не предусматривает выборку данных при выполнении команды TRSM(045).

Не производите изменение состояния бита запуска выборки данных (A50815) из программы. Состояние данного бита должно изменяться только с помощью Периферийного устройства.

**Пример**

Следующий ниже пример полностью описывает выполнение отслеживания данных.

Управляется с помощью Периферийного устройства. (Бит запуска выборки данных: A50815).

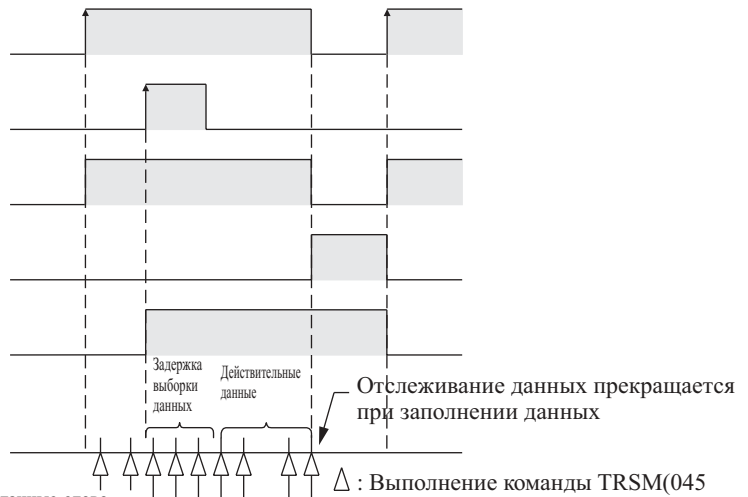
Бит запуска отслеживания данных: A50814.

Бит отслеживания данных: A50813

Бит завершения отслеживания данных: A50812

Флаг монитора запуска отслеживания данных: A50811

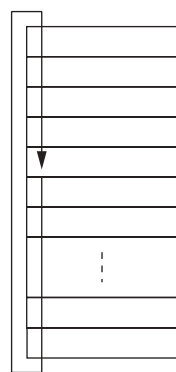
Выборка данных



Пример: данные-слово

Память отслеживания данных

См. Примечание



Данные действительны, начиная с данной точки

**Примечание:** Память отслеживания данных имеет кольцевую структуру. Данные записываются до конца памяти и, затем, запись производится с начала области, заканчиваясь как раз перед первой действительной выборкой данных.

## 3-28 Команды диагностики отказов

В настоящем разделе приводится описание команд, применяемых для определения и обработки ошибок.

Команда	Мнемоническое обозначение	Функциональный код	Страница
FAILURE ALARM	FAL	006	656
SEVERE FAILURE ALARM	FALS	007	659
FAILURE POINT DETECTION	FPD	269	662

## 3-28-1 Команда допускаемой ошибки FAILURE ALARM: FAL(006)

**Назначение**

По команде FAL(006) осуществляется генерация или сброс допускаемой ошибки. Определение допускаемой ошибки не прерывает работу Программируемого контроллера.

**Символ релейно-контактной схемы**

FAL(006)	
N	N: Номер FAL
M	M: Первое слово сообщения

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	FAL(006)
	Выполняется один раз для дифференцирования вверх	@ FAL(006)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Примечание:** Метод ввода номера *N* ошибки FAL различается для СХ-программатора и Пульта программирования. В случае использования СХ-программатора вводите номера от #0 до #511, и вводите номера от 000 до 501 при использовании Пульта программирования.

**Операнды**

В следующей ниже таблице показаны функции операндов.

N	M	Функция
#0000	#0001...#01FF	Осуществляет сброс допускаемой ошибки, имеющей соответствующий номер FAL.
	#FFFF	Осуществляет сброс всех допускаемых ошибок.
	Прочие*	Осуществляет сброс наиболее серьезной допускаемой ошибки.
От 0 до 511	#0000...#FFFF	Генерирует допускаемую ошибку с соответствующим номером FAL, (сообщение не передается).
	Адрес слова	Генерирует допускаемую ошибку с соответствующим номером FAL. Сообщение, содержащееся в словах от M до M+7 и состоящее из 16 символов ASCII кода, выводится на дисплей Периферийного устройства.

**Примечание:** Прочими установками являются константы от #0200 до #FFFF или адреса слов.

**Спецификации операндов**

Область	N	M
Область ввода/вывода (область CIO)	—	CIO 0000...CIO 6143
Рабочая область	—	W000...W511
Область удержания бита (Holding bit area)	—	H000...H511
Область вспомогательных битов	—	A000...A447 A448...A959
Область таймера	—	T0000...T4095

Область	N	M
Область счетчика	–	C0000...C4095
Область DM	–	D00000...D32767
Область EM, не содержащая банков	–	E00000...E32767
Область EM, содержащая банки	–	En_00000...En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	–	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)
Константы	Только указанные значения	#0000...#FFFF
Регистры данных	–	
Индексные регистры	–	
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15	

### Описание

Функционирование команды FAL(006) зависит от значения N. Для сброса всех ошибок устанавливайте N в значение, равное 0000. Устанавливайте N в значения от 0001 до 01FF для генерации ошибки.

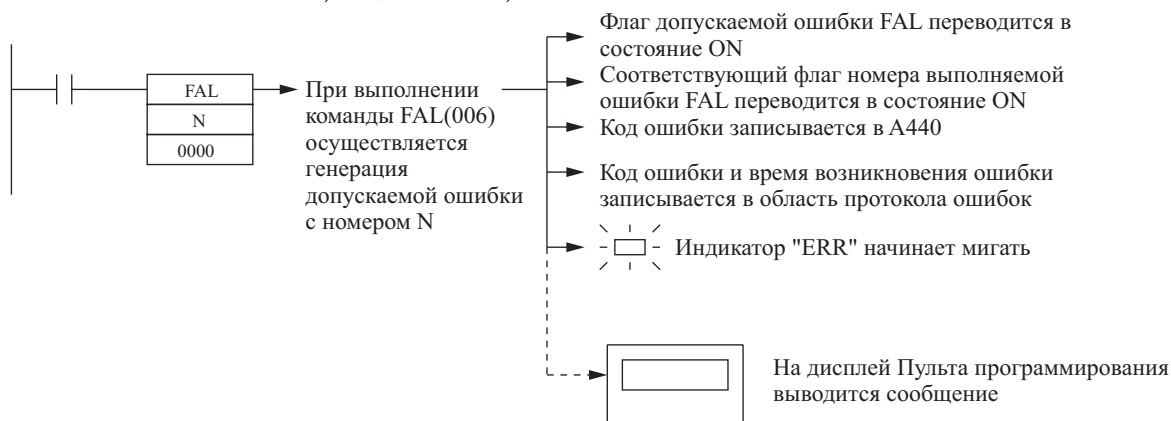
#### Генерация допускаемой ошибки

Когда команда FAL(006) выполняется при установке N в значение, соответствующее номеру FAL от &1 до &511, генерируется ошибка с заданным номером и выполняются следующие операции:

- 1,2,3...**
1. Флаг допускаемой ошибки (A40215) переводится в состояние ON. (Работа Программируемого контроллера продолжается.)
  2. Флаг номера выполняемой ошибки FAL переводится в состояние ON. Флаги A36001...A39115 соответствуют номерам FAL от 0001 до 01FF (от 1 до 511).
  3. Код ошибки записывается в A440. Коды ошибок от 4101 до 42FF соответствуют номерам FAL от 0001 до 01FF (1...511).

**Примечание:** При возникновении критической ошибки или более серьезной допускаемой ошибки в процессе выполнения команды FAL(006), код более серьезной ошибки записывается в A440.

4. Код ошибки и время возникновения ошибки записываются в область протокола ошибок (от A100 до A199).
5. Индикатор ошибки на Модуле центрального процессора начинает мигать.
6. Если в слове M указан адрес, на дисплей Периферийного устройства выводится сообщение, начинающееся в слове M.



В следующей ниже таблице для команды FAL(006) приводятся коды ошибок и флаги ошибок FAL.

Номер FAL	Коды ошибок FAL	Флаги выполняемой команды FAL
1...511 в десятичном коде	4101...42 FF	A36001...A39115



Если М является адресом слова и в М сохраняется сообщение, при выполнении команды FAL(006) это сообщение будет выводиться на дисплей Периферийного устройства. (Если вывод сообщения не требуется, задавайте в слове М константу.)

При выполнении команды FAL(006) производится регистрация сообщения, начинающегося в М. После регистрации сообщения, оно выводится на дисплей Пульта программирования, при его подключении.

В словах М...М+7 может храниться сообщение в ASCII коде длиной до 16 символов. При выводе на дисплей в первую очередь выводится содержание старшего байта каждого слова.

Кодом окончания сообщения является ноль (00 шестн.). Все 16 символов в словах М М+7 выводятся на дисплей, при этом символ «ноль» опускается.

Если содержание слов, в которых храниться сообщение, после выполнения команды FAL(006) изменяется, сообщение соответственно изменяется.

#### **Сброс допускаемых ошибок**

Когда команда FAL(006) выполняется при установке N в значение, равное 0, осуществляется сброс отдельной допускаемой ошибки или сброс всех допускаемых ошибок. (Сброс ошибок может также выполняться с помощью Периферийного устройства.)

Когда слово М устанавливается в значения от #0001 до #01FF(1...511), по команде FAL(006) производится сброс допускаемой ошибки с заданным номером. Соответствующий флаг номера выполняемой ошибки переводится в состояние OFF, а код ошибки в A440 стирается.

Когда значение М равно #FFFF, по команде FAL(006) производится сброс всех допускаемых ошибок, включая допускаемые системные ошибки.

Когда значение М находится в пределах от #2000 до #FFFF или является адресом слова, по команде FAL(006) производится сброс наиболее серьезной допускаемой ошибки, включая допускаемые системные ошибки.

При возникновении более чем одной ошибки FAL, ошибка с наименьшим номером сбрасывается.

#### **Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение N выходит за пределы значений от 0 до 511. OFF в других случаях.

В следующей ниже таблице показаны относящиеся к команде биты и флаги вспомогательной области.

Наименование	Адрес	Содержание
Флаг ошибки FAL (FAL Error Flag)	A40215	Переводится в состояние On, когда ошибка генерируется по команде FAL(006).
Флаги номера выполняемой ошибки (Executed FAL Number Flags)	A36001...A39115	Когда по команде FAL(006) генерируется ошибка, соответствующий флаг переводится в состояние ON. Флаги A36001...A39115 соответствуют номерам ошибок FAL 0001...1FF.
Область протокола ошибок	A100...A199	Область протокола ошибок содержит коды ошибок и время/дату их возникновения. Данные сохраняются для последних двадцати ошибок, включая ошибки, генерируемые командой FAL(006).
Код ошибки	A400	При возникновении ошибки, код этой ошибки сохраняется в A400. Кодами ошибок FAL от 0001 до 01FF являются соответственно 4101 42FF, При одновременном возникновении двух и более ошибок FAL, в A400 заносится код наиболее серьезной ошибки.

#### **Меры предосторожности**

Значение N должно находиться в пределах от 0000 до 01FF. В случае, когда значение N выходит за указанные пределы, определяется ошибка и флаг ошибки переводится в состояние ON.

#### **Примеры**

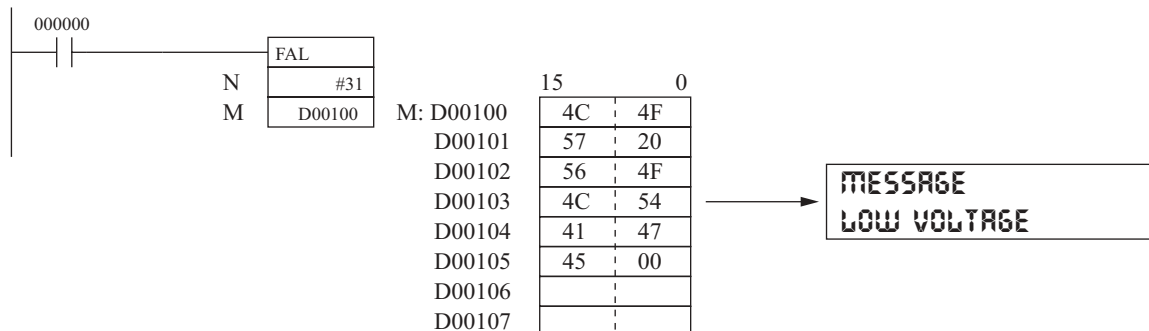
##### **Генерирование допускаемой ошибки**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, по команде FAL(006) генерируется допускаемая ошибка (номер FAL – 31) и выполняются следующие действия.

- 1,2,3... 1. Флаг ошибки FAL (A40215) переводится в состояние ON.
2. Соответствующий флаг номера выполняемой ошибки (A36114) переводится состояние ON.
3. Соответствующий код ошибки (411F) заносится в A400.

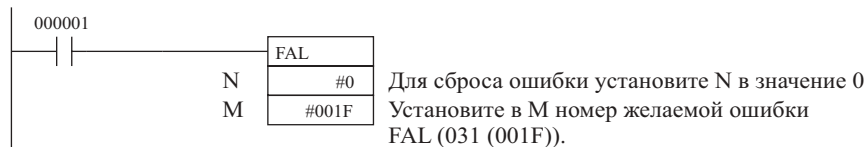
**Примечание:**

- При одновременном возникновении двух и более ошибок FAL, в A400 заносится код наиболее серьезной ошибки (код ошибки, имеющий больший номер).
4. Код ошибки и время/дата возникновения ошибки заносится в область протокола ошибок (A100...A199).
  5. Индикатор «ERR» Модуля центрального процессора начинает мигать.
  6. Сообщение в ASCII коде, находящееся в D00100...D00107 выводится на дисплей Периферийного устройства. (В случае если сообщение не требуется,



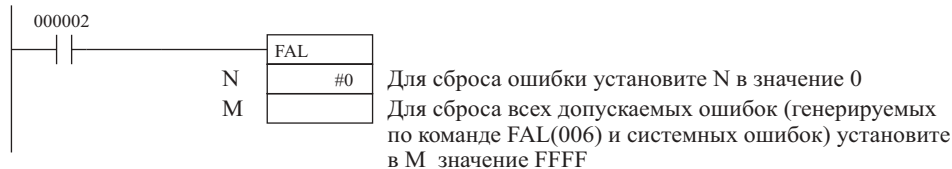
#### Сброс отдельной допускаемой ошибки

В следующем ниже примере, когда СЮ 000001 находится в состоянии ON, по команде FAL(006) осуществляется сброс допускаемой ошибки FAL с номером 31, перевод в состояние OFF флага номера выполняемой ошибки (A36114), и флага допускаемой ошибки FAL (A40215).



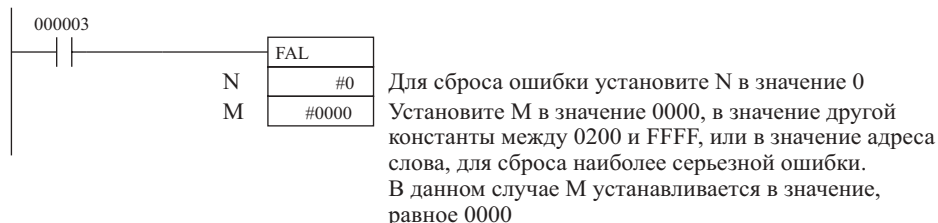
#### Сброс всех допускаемых ошибок

В следующем ниже примере, когда СЮ 000002 находится в состоянии ON, по команде FAL(006) осуществляется сброс всех допускаемых ошибок FAL, перевод в состояние OFF флагов номера выполняемых ошибок (A36001...A39115), и флага допускаемой ошибки FAL (A40215).



#### Сброс наиболее серьезной допускаемой ошибки

В следующем ниже примере, когда СЮ 000003 находится в состоянии ON, по команде FAL(006) осуществляется сброс наиболее серьезной допускаемой ошибки FAL и сброс кода ошибки в A400. Если данная ошибка генерировалась по команде FAL(006), соответствующий флаг номера выполняемой ошибки (A36114), и флаг допускаемой ошибки FAL (A40215) переводятся в состояние OFF.



### 3-28-2 Команда критической ошибки SEVERE FAILURE ALARM: FALS(007)

#### Назначение

Команда FALS(007) генерирует критическую ошибку. При появлении критической ошибки работа Программируемого контроллера прерывается.

**Символ релейно-контактной схемы**

—	FALS(007)	
	N	N: Номер критической ошибки FALS
	M	M: Первое слово сообщения

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	FALS(007)
	Выполняется один раз для дифференцирования вверх	@ FALS(007)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****N: Номер FALS**

Указывает номер генерируемой ошибки FALS. Значение N должно находиться в пределах от 1 до 511. Команды FALS(006) и FALS(007) используют одни и те же номера FALS.

**M: Первое слово сообщения.**

Указывает первое из восьми слов сообщения в ASCII коде, которое должно выводиться на дисплей периферийного устройства при выполнении команды FALS(007).

В случае, когда сообщение не требуется, указывайте константу (0000...FFFF).

**Спецификации операндов**

Область	N	M
Область ввода/вывода (область CIO)	—	CIO 0000...CIO 6143
Рабочая область	—	W000...W511
Область удержания бита (Holding bit area)	—	H000...H511
Область вспомогательных битов	—	A000...A959
Область таймера	—	T0000...T4095
Область счетчика	—	C0000...C4095
Область DM	—	D00000...D32767
Область EM, не содержащая банков	—	E00000...E32767
Область EM, содержащая банки	—	En_00000...En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	—	@D00000...@D32767 @E00000...@E32767@ En_00000 @En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)
Константы	Только указанные значения	#0000...#FFFF
Регистры данных	—	—
Индексные регистры	—	—
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(++), ,-(-)IR0...,-(-)IR15	

**Описание**

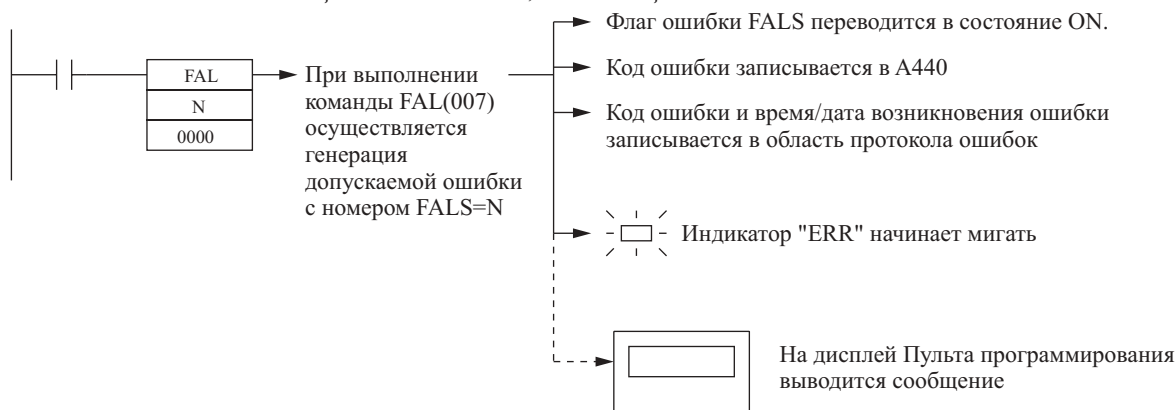
При выполнении команды FALS(007) генерируется критическая ошибка с указанным номером FALS, и выполняются следующие действия:

- 1,2,3...** 1. Флаг критической ошибки (A40106) переводится в состояние ON. (Работа Программируемого контроллера прерывается.)

2. Код ошибки записывается в A440. Коды ошибок от C101 до C2FF соответствуют номерам FALS от 0001 до 01FF (1...511).

**Примечание:** Если возникает более серьезная ошибка, нежели ошибка по команде FALS(007) (т.е. ошибка, имеющая более высокий код), A400 будет содержать код более серьезной ошибки.

3. Код ошибки и время/дата возникновения ошибки записываются в область протокола ошибок (от A100 до A199).
4. Индикатор ошибки на Модуле центрального процессора начинает мигать.
5. Если в слове M указан адрес, на дисплей Периферийного устройства выводится сообщение в ASCII коде, начинающееся в слове M.



В следующей ниже таблице для команды FALS(007) приводятся коды ошибок.

Номер FALS	Коды ошибок FALS
1...511	C101...C2 FF

**Примечание:** Метод ввода номера N ошибки FALS различается в случае применения CX-программатора и применения Пульта программирования. При использовании CX-программатора вводите номера от #0 до #511, и вводите номера от 000 до 501 при использовании Пульта программирования.

#### Дисплей сообщений

В случае, когда слово N установлено в значение, равное 0000, при выполнении команды FALS(007) может производиться сброс отдельной критической ошибки или сброс всех критических ошибок. (Сброс ошибок можно также производить помощью Периферийного устройства.)

Если M является адресом слова, при выполнении команды FALS(007) на дисплей Периферийного устройства будет выводиться сообщение, начинающееся в M. (Если вывод сообщения не требуется, задавайте в слове M константу.)

При выполнении команды FALS(007) производится регистрация сообщения, начинающегося в M. После регистрации сообщения, оно выводится на дисплей Пульта программирования после его подключения. В словах M...M+7 может храниться сообщение в ASCII коде длиной до 16 символов. При выводе на дисплей в первую очередь выводится содержание старшего байта каждого слова.

Кодом окончания сообщения является ноль (00 шестн.). Все 16 символов в словах M M+7 выводятся на дисплей, при этом символ «ноль» опускается.

Если содержание слов, в которых храниться сообщение, после выполнения команды FALS(007) изменяется, сообщение соответственно изменяется.

#### Сброс ошибок FALS(007)

Перед сбросом ошибок, генерируемых по команде FALS(007), вначале устраните причину возникновения ошибки, затем либо выполните сброс из Пульта программирования, либо кратковременно отключите питание Программируемого контроллера.

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение N выходит за пределы значений от 0001 до 01FF (1...511). OFF в других случаях.

В следующей ниже таблице показаны относящиеся к команде биты и флаги вспомогательной области.

Наименование	Адрес	Содержание
Флаг ошибки FALS	A40106	Переводится в состояние ON, когда ошибка генерируется по команде FALS(007).
Область протокола ошибок	A100...A199	Область протокола ошибок содержит коды ошибок и время/дату их возникновения. Данные сохраняются для последних двадцати ошибок, включая ошибки, генерируемые командой FALS(007).
Код ошибки	A400	При возникновении ошибки, код этой ошибки сохраняется в A400. Кодами ошибок FALS от 0001 до 01FF являются соответственно C101 C2FF. При одновременном возникновении двух и более ошибок, в A400 заносится код наиболее серьезной ошибки.

### Меры предосторожности

Кодом окончания сообщения является ноль (00 шестн.). Все 16 символов в словах M M+7 выводятся на дисплей, при этом символ «ноль» опускается.

Значение N должно находиться в пределах от 0001 до 01FF. В случае, когда значение N выходит за указанные пределы, определяется ошибка и флаг ошибки переводится в состояние ON.

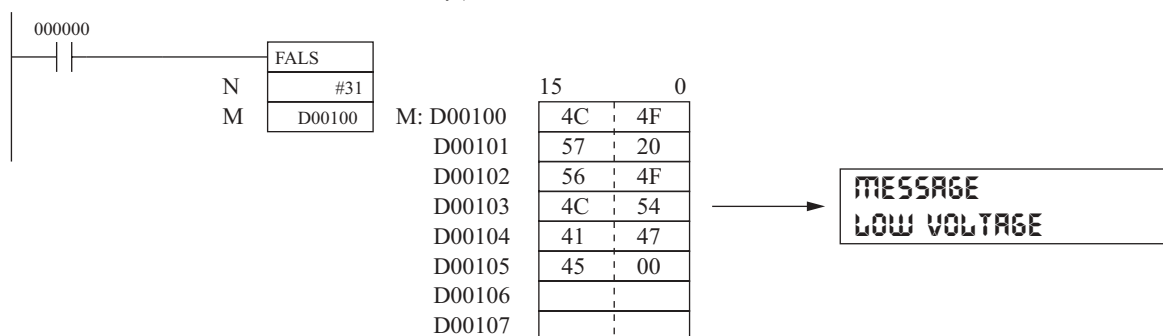
### Примеры

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, по команде FALS(007) генерируется критическая ошибка (номер FALS – 31) и выполняются следующие действия.

- 1,2,3... 1. Флаг ошибки FALS (A40106) переводится в состояние ON.
2. Соответствующий код ошибки (C11F) заносится в A400.

**Примечание:** A400 содержит код наиболее серьезной из всех произошедших ошибок, включая допускаемые ошибки и ошибки системы, а также ошибки, генерируемые по командам FAL и FALS. При одновременном возникновении двух и более ошибок FAL, в A400 заносится код наиболее серьезной ошибки (код ошибки, имеющий больший номер).

3. Код ошибки и время/дата возникновения ошибки заносится в область протокола ошибок (A100...A199).
4. Индикатор «ERR» Модуля центрального процессора начинает мигать.
5. Сообщение в ASCII коде, находящееся в D00100...D00107 выводится на дисплей Периферийного устройства. (В случае если сообщение не требуется, указываете для M константу.)

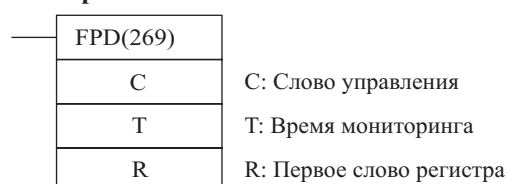


### 3-28-3 Команда определения точки появления ошибки FAILURE POINT DETECTION: FPD(269)

#### Назначение

Команда FPD(269) осуществляет диагностику ошибки посредством мониторинга времени между выполнением команды FPD(369) и работой диагностического вывода, определяя ввод, препятствующий переводу вывода в состояние ON.

#### Символ релейно-контактной схемы



**Модификации**

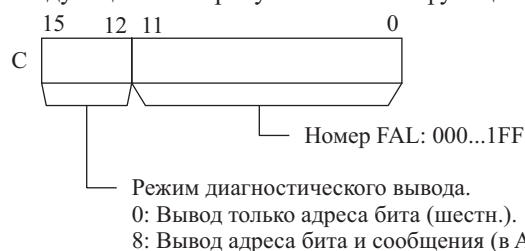
Модификации	Выполняется в каждом цикле для состояния ON	FPD(269)
	Выполняется один раз для дифференцирования вверх	@ FPD(269)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Не применяются	Да	Да	Не применяются

**Операнды****С: Слово управления**

Слово С должно быть константой в диапазоне значений от 0000 до 01FF или от 8000 до 81FF. На следующем ниже рисунке показано функциональное назначение цифр слова управления.

**T: Время мониторинга**

Значение слова T должно находиться между 0000 и 270F (0 9999 в десятичном коде). Значение, равное 0, отменяет мониторинг. Значения от 1 до 270F соответствуют величинам от 0.1 до 999.9 сек.

**R: Первое слово регистра**

Функции этого слова описаны ниже.

**Спецификации операндов**

Область	C	T	R
Область ввода/вывода (область CIO)	—	CIO 0000...CIO 6143	
Рабочая область	—	W000...W511	
Область удержания бита (Holding bit area)	—	H000...H511	
Область вспомогательных битов	—	A000...A447 A448...A959	A448...A959
Область таймера	—	T0000...T4095	
Область счетчика	—	C0000...C4095	
Область DM	—	D00000...D32767	
b120Область EM, не содержащая банков	—	E00000...E32767	
Область EM, содержащая банки	—	En_00000...En_32767 (n=0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	—	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n=0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)	
Константы	Только указанные значения	#0000...#270F (двоичные)	—
Регистры данных	—		
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	—	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,(- -)IR0...(- -)IR15	

**Описание**

По команде FDP(269) выполняется мониторинг процесса и логическая диагностика. Функция хронометрированного мониторинга генерирует допустимую ошибку с указанным номером FAL, если диагности-



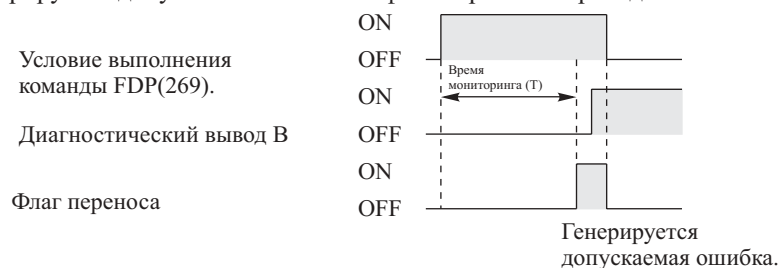
ческий выход не переводится в состояние ON в течение указанного времени. Функция логической диагностики указывает ввод, препятствующий переводу вывода в состояние ON.



**Примечание:** Блок логической диагностики начинается с первой команды LD (не LD TR) или команды LD NOT, расположенной после команды FDP(269) и заканчивается первой командой OUT (не OUT TR) или другой командой, расположенной справа.

#### Функция хронометрированного мониторинга

Команда FDP(269) начинает отсчет времени, когда условие выполнения А переводится в состояние ON. Если в течение контролируемого времени диагностический вывод не переводится в состояние ON, генерируется допустимая ошибка и флаг переноса переводится в состояние ON.



**Примечание:** Диагностический вывод должен быть переведен в состояние ON в течение контролируемого времени (T). Для автоматического установления времени мониторинга может использоваться обучающая функция.

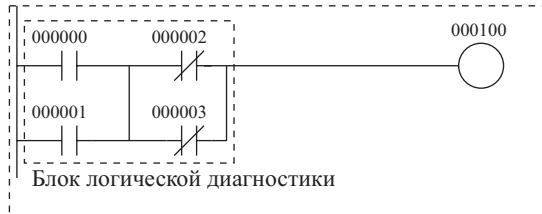
При переводе флага переноса в состояние ON выполняются следующие ниже действия. (Эти действия не выполняются, если в слове С номер FAL установлен в значение 000).

- 1,2,3... 1. Флаг ошибки FAL переводится в состояние ON. (Работа Программируемого контроллера продолжается.)
2. Флаг выполняемого номера FAL переводится в состояние ON. (Флаги A36001 A39115 соответствуют номерам FAL 0011FF.)
3. Код соответствующей ошибки записывается в A400. Коды ошибок 4101...42FF соответствуют номерам FAL 0011FF.
4. Код ошибки и время/дата возникновения ошибки заносится в область протокола ошибок (A100...A199).
5. Индикатор «ERR» Модуля центрального процессора начинает мигать.
6. Если режим вывода установлен в режим вывода адреса бита и сообщения (цифра в старшем разряде слова С установлена в значение 8), сообщение в ASCII коде, сохраняемое в словах R+6...R+9 будет выведено на дисплей в качестве сообщения о допустимой ошибке.

### Функция логической диагностики

В каждом из циклов, когда условие выполнения команды FDP(269) находится в состоянии ON, по команде FDP(269) производится определение вывода, препятствующего переводу диагностического вывода в состояние ON, и запись адреса этого бита в область регистра, начиная с R.

Если в следующем ниже примере биты ввода CIO 000000...CIO 000003 находятся в состоянии ON, по команде FDP(269) определяется, что нормально замкнутое условие CIO 000002 является причиной нахождения CIO 000010 в состоянии OFF. Команда FDP(269) переводит флаг нахождения адреса бита (бит 15 слова R) в состояние ON и записывает адрес бита в слова регистра R+2...R+4.



Условие выполнения логической диагностики

Функция логической диагностики выполняется в каждом из циклов до тех пор, пока условие выполнения команды FDP(269) находится в состоянии ON. Функция логической диагностики независима от функции хронометрированного мониторинга.

В случае, когда два и более битов препятствуют переводу диагностического вывода в состояние ON, в слова R+2...R+4 заносится адрес первого бита в условии выполнения (на верхней командной линии и ближе к левой линии шины).

Биты ввода в командах LD, LD NOT, AND, AND NOT, OR, OR NOT (включая дифференцированные модификации и модификации с немедленной регенерацией) проверяются функцией логической диагностики.

Биты ввода в других командах и операнды, указываемые косвенным методом через индексные регистры, не проверяются.

Блок логической диагностики начинается с первой команды LD (не LD TR) или команды LD NOT, расположенной после команды FDP(269), и заканчивается первой командой OUT (не OUT TR) или другой командой, расположенной справа.

Существует два режима работы диагностического вывода, задаваемые значением цифры старшего разряда слова C.

- 1,2,3... 1. Режим вывода адреса бита (цифра старшего разряда в слове C равна 0).  
 Бит 15 слова R (флаг нахождения адреса бита) переводится в состояние ON, когда адрес бита ввода найден, и бит 14 слова R обозначает нормальное состояние ввода (ON или OFF).  
 Адрес памяти внутреннего ввода/вывода, принадлежащий биту ввода, заносится в слова R+2...R+4 в виде восьмизначного шестнадцатеричного числа.
2. Режим вывода адреса бита и сообщения (цифра старшего разряда в слове C равна 8).  
 Бит 15 слова R (флаг нахождения адреса бита) переводится в состояние ON, когда адрес бита ввода найден, и бит 14 слова R обозначает нормальное состояние ввода (ON или OFF).  
 Адрес бита ввода, заносится в слова R+2...R+4 в виде шести символов ASCII кода.

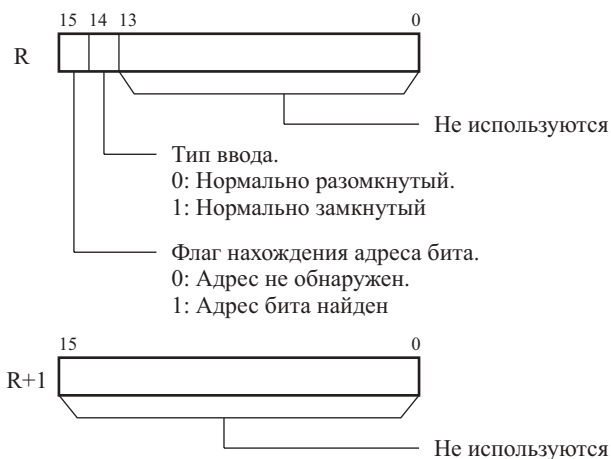
### Функции слов регистра

Слова регистра содержат результаты диагностики и могут также содержать сообщение об ошибке в ASCII коде, которое выводится на дисплей в том случае, когда функцией хронометрированного мониторинга генерируется ошибка. Функции слов регистра зависят от режима работы вывода диагностики, устанавливаемого значением цифры старшего разряда в слове C.

#### Вывод адреса бита (C = 0 \_ \_ \_)

Когда цифра старшего разряда в слове C устанавливается в значение 0, адрес памяти внутреннего ввода/вывода, принадлежащий биту ввода, заносится в слова R+2 и R+3 в виде восьмизначного шестнадцатеричного числа. R содержит два флага, указывающие на обнаружение бита ввода, и на использование этого бита в качестве нормально разомкнутого или нормально замкнутого условия ввода.

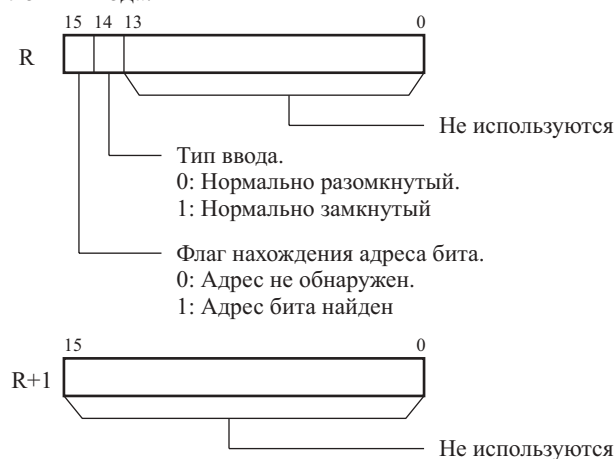




R+2  R+3

#### Вывод адреса бита и сообщения (C = 8 \_ \_ \_)

Когда цифра старшего разряда в слове C устанавливается в значение 8, адрес бита ввода, заносится в слова R+2 и R+3 в виде шести символов ASCII кода. R содержит два флага, указывающие на обнаружение бита ввода, и на использование этого бита в качестве нормально разомкнутого или нормально замкнутого условия ввода.



Слова регистра R+2...R+4 обозначают адрес ввода, который препятствует переводу диагностического вывода в состояние ON. Адрес бита выводится в эти слова в коде ASCII. В следующей ниже таблице показано использование кода ASCII в каждой из областей памяти.

Область	Текст в ASCII коде	Примечание
Вспомогательная область	A00000...A95915	—
Область удержания	H00000...H51115	—
Рабочая область	W00000...W51115	—
Область СЮ	000000...665515	—
Область задач	TK0000...TK0031	—
Область таймера	_T0000..._T4095	«_» является пробелом в ASCII I коде. (Код символа 20.)
Область счетчика	_C0000..._C4095	

	15	
R+2	W	5
R+3	1	1
R+4	1	5

Адрес бита записывается в ASCII коде

Слова регистра R+2...R+5 для W51115 имеют следующие значения.

Слово	Биты 8...15	Биты 0...7
R + 2	W	5
R + 3	1	1
R + 4	1	5
R + 5	2D (шестн.)	Тип ввода (шестн.). 30: Нормально разомкнутый. 31: Нормально замкнутый.

В словах R+6...R+9 пользователь может сохранить сообщение в ASCII коде. Это сообщение будет выведено на дисплей Периферийного устройства в случае, когда допускаемая ошибка генерируется функцией хронометрированного мониторинга. Непременно обозначайте конец сообщения символом 0 (00 шестн.).

	15	8	7	0
R+6				
R+7				
R+8				
R+9				

### Задание времени мониторинга с помощью функции обучения

В случае, когда для T задается адрес слова, время мониторинга может устанавливаться автоматически, при помощи функции обучения. Если для T задается адрес бита, используйте следующую ниже процедуру.

- Примечание:**
1. Переведите в состояние ON бит обучения FPD (A59800).
  2. По команде FPD(269) производится измерение времени от точки, в которой условие выполнения команды переводится в состояние ON, до перевода диагностического вывода в состояние ON.
  3. Если измеренное время превышает значение, установленное для времени мониторинга, в T записывается значение, в полтора раза большее измеренной величины.

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение C выходит за пределы значений от 0000 до 01FF или от 8000 до 81 FF. Переводится в состояние ON, когда значение T выходит за пределы допускаемых значений от 0000 до 270F. OFF в других случаях.
Флаг переноса	CY	Переводится в состояние ON, когда диагностический вывод остается в состоянии OFF после истечения заданного времени мониторинга.

В следующей ниже таблице показаны относящиеся к команде биты и флаги вспомогательной области.

Наименование	Адрес	Содержание
Флаг ошибки FAL	A40215	Переводится в состояние On, если в течение времени мониторинга регистрируется допускаемая ошибка FAL.
Флаги номера выполняемой ошибки	A36001...A39115	Когда в течение времени мониторинга регистрируется допускаемая ошибка FAL, соответствующий флаг переводится в состояние ON. Флаги A36001...A39115 соответствуют номерам ошибок 0001...-1FF.
Область протокола ошибок	A100...A199	Область протокола ошибок содержит коды ошибок и время/дату возникновения ошибок. Данные сохраняются для последних двадцати ошибок, включая ошибки, генерируемые командой FPD(236).
Код ошибки	A400	При возникновении ошибки, код этой ошибки сохраняется в A400. Кодами ошибок FAL от 0001 до 01FF являются соответственно 4101 42FF, При одновременном возникновении двух и более ошибок, в A400 заносится код наиболее серьезной ошибки.
Бит обучения FPD	A59800	Если вы желаете, чтобы время мониторинга было установлено автоматически, (с помощью бита обучения), при выполнении команды FPD(269) переведите данный бит в состояние ON.

### Меры предосторожности

В случае использования функции хронометрированного мониторинга условие выполнения для команды FPD(269) должно оставаться в состоянии ON в течение всего времени мониторинга, установленного в слове T.

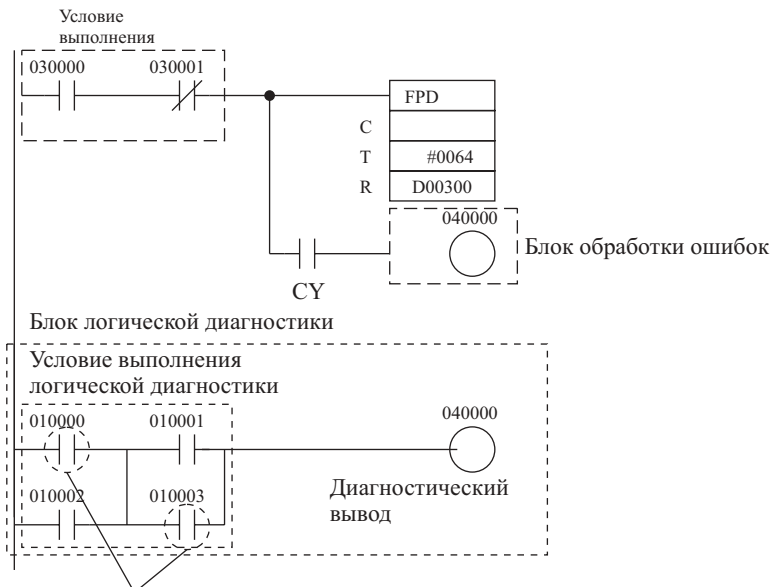
Условие выполнения для команды FPD(269) должно составляться из комбинации нормально разомкнутого и нормально замкнутого ввода.

Блок обработки ошибок поставляется по отдельному заказу. Если блок обработки ошибок присутствует в программе, непременно используйте выходы или другие правосторонние команды. В этой точке команды LD и LD NOT использоваться не могут.

Команда FPD(269) может использоваться в программе неоднократно, однако каждая из команд должна иметь свою собственную установку в регистре (R).

### Примеры

Следующий пример программы демонстрирует работу функции хронометрированного мониторинга и функции логической диагностики. В данном примере диагностический вывод CIO 02000 остается в состоянии OF вследствие того, что CIO 010000 и CIO 010003 остаются в состоянии OFF в условии выполнения логической диагностики.



Диагностический вывод функции логической диагностики (CIO 020000) остается в состоянии OFF вследствие того, что эти условия ввода находятся в состоянии OFF.

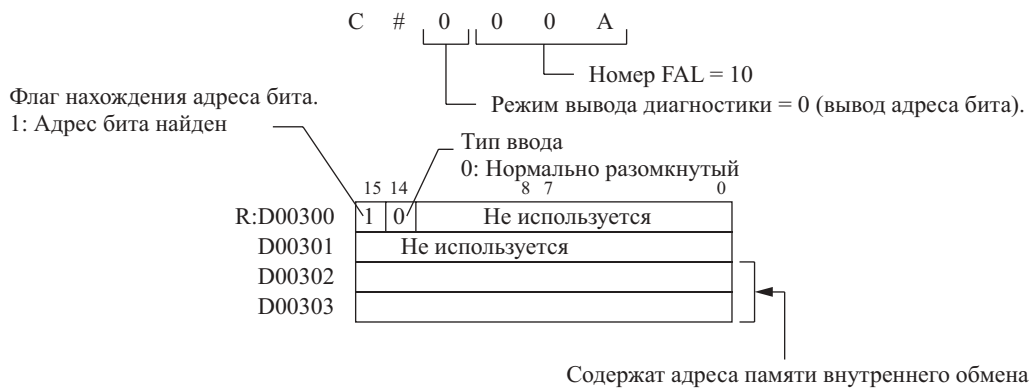
### Функция логической диагностики

Когда диагностический вывод (CIO 020000) остается в состоянии OFF в течение 10 секунд после того, как CIO 030000 и CIO 030001 находятся в состоянии ON, генерируется допускаемая ошибка и выполняются следующие ниже операции.

- 1,2,3... 1. Флаг переноса переводится в состояние ON.
2. Когда три цифры младших разрядов слова C указывают номер FAL 00A (шестн.), флаг выполняемой ошибки FAL (A36010) переводится в состояние ON, соответствующий код ошибки (410A) записывается в A400, и флаг ошибки FAL (A40215) переводится в состояние ON.

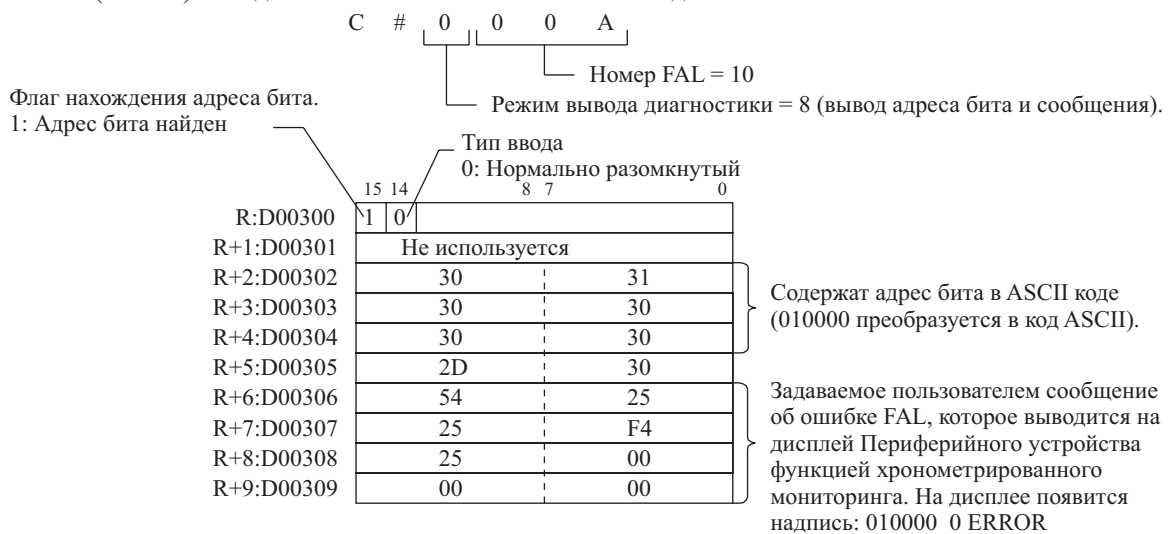
### Функция логической диагностики (C=000A)

Так как цифра в старшем разряде слова C равна нулю (режим вывода адреса бита) адрес памяти внутреннего ввода/вывода CIO 010000 выводится в D00303 и D00302. (CIO 010000 находится в более высокой командной линии, чем CIO 010003.)



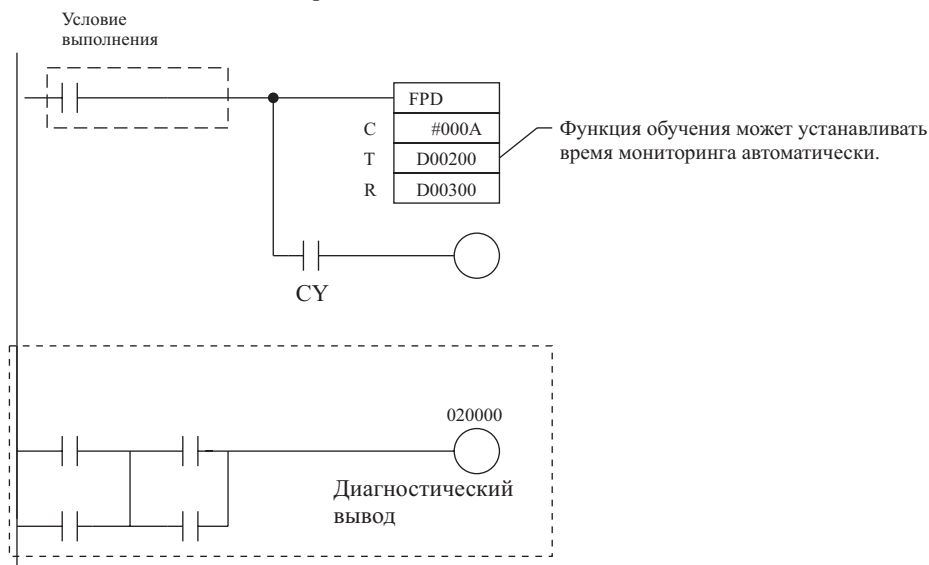
**Функция логической диагностики (C=800A)**

Так как цифра в старшем разряде слова C равна 8 (режим вывода адреса бита и сообщения) адрес CIO 010000 (010000) выводится в D00302...D00304 в ASCII коде.



**Задание времени мониторинга с помощью функции обучения**

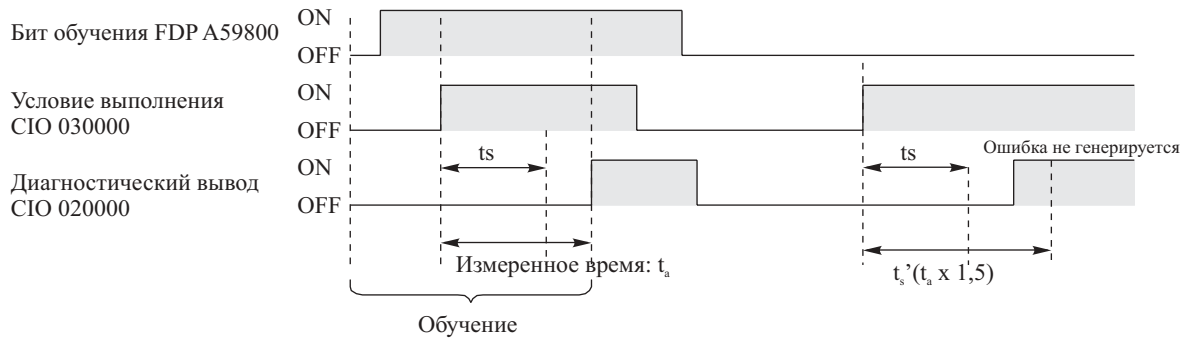
Время мониторинга устанавливается автоматически, при помощи функции обучения, когда в качестве значения для T задается адрес слова.



Для запуска функции обучения переведите флаг A59800 (флаг функции обучения) в состояние ON. Когда флаг A59800 переводится в состояние ON, по команде FPD(269) производится измерение времени от точ-

ки, в которой условие выполнения (CIO 030000) переводится в состояние ON, до перевода диагностического вывода (CIO 020000) в состояние ON.

Если измеренное время превышает значение, установленное для времени мониторинга, в качестве нового значения времени мониторинга в Т записывается время, в полтора раза большее измеренной величины.



ts: Начальная установка времени мониторинга Т.

ta: Измеренное время.

ts': Новое значение времени мониторинга.

(Когда  $t_a > t_s$ ,  $t_s' = t_a \times 1,5$ )

### 3-29 Прочие команды

В настоящем разделе приводится описание команд, при помощи которых осуществляются операции с флагом переноса, выбор EM банка и расширение максимальной длительности цикла.

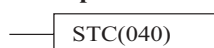
Команда	Мнемоническое обозначение	Функциональный код	Страница
SET CARRY	STC	040	671
CLEAR CARRY	CLC	041	671
SELECT EM BANK	EMBC	281	672
EXTEND MAXIMUM CYCLE TIME	WDT	094	674

#### 3-29-1 Команда установки флага переноса SET CARRY: STC(040)

##### Назначение

Данная команда устанавливает флаг переноса (CY).

##### Символ релейно-контактной схемы



##### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	STC(040)
	Выполняется один раз для дифференцирования вверх	@ STC(040)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

##### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

##### Описание

Когда условие выполнения находится в состоянии ON, по команде STC(040) производится перевод флага переноса (CY) в состояние ON. Хотя команда STC(040) изменяет состояние флага переноса, этот флаг будет изменять свое состояние при выполнении последующих команд, влияющих на его состояние.

##### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	OFF
Флаг переноса	CY	ON
Флаг отрицательного значения	N	OFF

##### Меры предосторожности

Команды ROL(027), ROLL(572), ROR(028) и RORL(573) могут использовать флаг переноса при выполнении операций вращения. При выполнении этих команд используйте команды STC(040) и CLC(041) для установки и сброса флага переноса.

#### 3-29-2 Команда очистки (сброса) флага переноса CLEAR CARRY: CLC(041)

##### Назначение

Данная команда устанавливает флаг переноса (CY) в состояние OFF.

##### Символ релейно-контактной схемы



##### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	CLC(041)
	Выполняется один раз для дифференцирования вверх	@ CLC(041)
	Выполняется один раз для дифференцирования вниз	Не поддерживается

Указание немедленной регенерации	Не поддерживается
----------------------------------	-------------------

### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

### Описание

Когда условие выполнения находится в состоянии ON, по команде CLC(041) производится перевод флага переноса (CY) в состояние OFF. Хотя команда CLC(041) изменяет состояние флага переноса, этот флаг будет изменять свое состояние при выполнении последующих команд, влияющих на его состояние.

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF
Флаг равенства	=	OFF
Флаг переноса	CY	OFF
Флаг отрицательного значения	N	OFF

### Меры предосторожности

Команды +C(402), +CL(403), +BC(406) и +BCL(407) могут использовать флаг переноса при выполнении операций суммирования. С целью предотвращения влияния других предшествующих команд, вводите команду CLC(041) непосредственно перед любой из перечисленных выше команд.

Команды -C(412), -CL(413), -BC(416) и -BCL(417) могут использовать флаг переноса при выполнении операций вычитания. С целью предотвращения влияния других предшествующих команд, вводите команду CLC(041) непосредственно перед любой из перечисленных выше команд.

Команды ROL(027), ROLL(572), ROR(028) и RORL(573) могут использовать флаг переноса при выполнении операций вращения. При выполнении этих команд используйте команды STC(040) и CLC(041) для установки и сброса флага переноса.

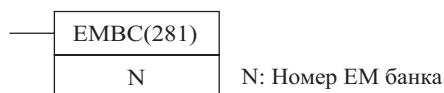
***Примечание:** Команды +(400), +L(401), +B(405), -(410), -L(411), -B(414) и -BL(415) RORL(573) не используют флаг переноса при выполнении операций суммирования и вычитания. В общем случае применяйте данные команды для выполнения операций суммирования и вычитания.*

### 3-29-3 Команда выбора EM банка SELECT EM BANK: EMBC(281)

#### Назначение

Данная команда производит изменение текущего банка.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	EMBC(281)
	Выполняется один раз для дифференцирования вверх	@ EMBC(281)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

### Операнды

#### N: Номер EM банка

Указывает новый номер EM банка в шестнадцатеричном коде (0000 000C).

## Спецификации операндов

Область	N
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143
Рабочая область	W000...W511
Область удержания бита (Holding bit area)	H000...H511
Область вспомогательных битов	A000...A959
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область DM	D00000...D32767
Область EM, не содержащая банков	E00000...E32767
Область EM, содержащая банки	En_00000...En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767@ En_00000...@En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)
Константы	#0000...#000C (двоичные)
Регистры данных	DR0...DR15
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,(- -)IR0...,-(- -)IR15

## Описание

По команде EMBC(281) производится изменение текущего EM банка (расширенной памяти данных) на банк, указываемый значением N. В то же время номер нового банка заносится в A301.

В области EM доступно 13 банков (от 0 до C), каждый из банков содержит 32768 слов (от E00000 до E32767). Адреса EM памяти могут указываться двумя способами. При использовании первого способа для изменения текущего банка должна применяться команда EMBC(281).

- 1,2,3...**
1. Для указания адресов в текущем EM банке адреса EM памяти могут указываться без номера банка, т.е. E00000...E32767 (n=0...C).
  2. Для обозначения адресов в определенном EM банке адреса могут указываться с номером банка, т.е. En\_00000...En\_32767 (n=0...C).

## Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда значение N выходит за пределы значений от 0000 до 000C. Переводится в состояние ON, когда N указывает несуществующий EM банк. (Эта ошибка возникает, когда указываемый EM банк регистрируется в начальных установках Программируемого контроллера в качестве памяти файлов) OFF в других случаях.

В следующей ниже таблице показаны относящиеся к команде флаги вспомогательной области.

Наименование	Адрес	Функционирование
Текущий EM банк	A301	Содержит номер текущего EM банка в шестнадцатеричном коде (0000...000C).

## Меры предосторожности

Номер текущего EM банка, изменяемый в циклической задаче, сохраняется при переключении работы между задачами. Например, если команда EMBC(281) применяется в задаче 1 для изменения текущего банка В на банк С, банк С остается текущим банком для всех циклических задач, даже если производится переход к выполнению задачи 2.

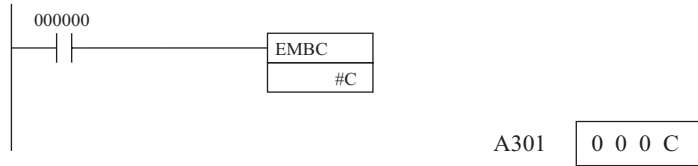
Номер текущего EM банка, изменяемый в задаче прерывания, сохраняется только на время выполнения прерывания, в котором изменялся номер банка. После завершения выполнения задачи прерывания возвращается номер предшествующего банка EM памяти.



В случае, когда указываемый банк зарегистрирован в начальных установках Программируемого контроллера в качестве памяти файлов, определяется ошибка.

### Примеры

В следующем ниже примере, когда СЮ 000000 переводится в состояние ON, номер текущего банка изменяется на С, и номер нового банка (000С шестн.) заносится в А301.

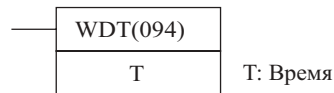


### 3-29-4 Команда увеличения максимальной длительности цикла EXTEND MAXIMUM CYCLE TIME: WDT(094)

#### Назначение

При выполнении данной команды производится увеличение максимальной длительности цикла, однако, только на один цикл, в котором выполняется команда. Команда WDT(094) может использоваться для предотвращения определения ошибки (превышения длительности цикла) в том случае, когда для выполнения специальных действий временно требуется увеличенная длительность цикла.

#### Символ релейно-контактной схемы



#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	WDT(094)
	Выполняется один раз для дифференцирования вверх	@ WDT(094)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

##### T: Время

Указывает установку для следящего таймера. Значение установки от 0000 до 0F9F (шестн.) или &0000...&3999 (десятичные).

#### Спецификации операндов

Область	T
Область ввода/вывода (область СЮ)	—
Рабочая область	—
Область удержания бита (Holding bit area)	—
Область вспомогательных битов	—
Область таймера	—
Область счетчика	—
Область DM	—
Область EM, не содержащая банков	—
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	0000...0F9F (двоичные)
Регистры данных	—

Область	T
Индексные регистры	–
Косвенная адресация с использованием индексных регистров	–

### Описание

По команде WDT(094) производится увеличение максимальной длительности цикла, в котором выполняется команда. Установка времени для следящего таймера в начальных установках Программируемого контроллера увеличивается на величину T...10 мсек. (0...39990 мсек.)

В следующей ниже таблице показаны установки времени для следящего таймера в начальных установках Программируемого контроллера. Значением максимальной длительности цикла по умолчанию является 1000 мсек. Диапазон изменения значений – от 1 до 40000 мсек. в единицах, равных 10 мсек.

Наименование	Функция	Значения параметра
Время для следящего таймера	Если длительность текущего цикла превышает заданную максимальную длительность цикла, определяется критическая ошибка превышения длительности цикла.	0: Значение по умолчанию (1000 мсек.) 1: Длительность, устанавливаемая пользователем.
	Устанавливает максимальную длительность цикла. (Данная установка действительна тогда, когда первоначально задано значение, равное 1.)	0001...0FA0 (1 40000 мсек. Единицы-10мсек.)

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда время, заданное для следящего таймера превышает значение, равное 40000 мсек. OFF в других случаях.

В следующей ниже таблице показаны относящиеся к команде флаги вспомогательной области.

Наименование	Адрес	Функционирование
Флаг превышения длительности цикла	A40108	Переводится в состояние ON, когда длительность текущего цикла превышает заданную максимальную длительность цикла (ожидаемую длительность цикла), заданную в начальных установках Программируемого контроллера. Данная ошибка является критической и приводит к прерыванию выполнения программы.
Максимальная длительность цикла	A262 и A263	Эти слова содержат значение максимальной длительности цикла в виде двоичного числа длиной 32 разряда. Данное значение обновляется в каждом из циклов.
Длительность текущего цикла	A264 и A265	Эти слова содержат значение длительности текущего цикла в виде двоичного числа длиной 32 разряда. Данное значение обновляется в каждом из циклов.

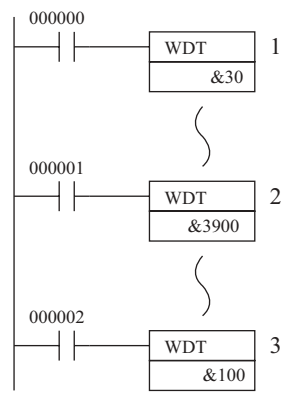
### Меры предосторожности

Команда WDT(094) может использоваться в одном цикле несколько раз. В этом случае увеличение длительности цикла, задаваемое каждой из команд, суммируется, однако суммарное время не должно превышать 40000 мсек. Команда WDT(094) не выполняется, если длительность цикла уже увеличена до значения, равного 40000 мсек.

### Примеры

В данном примере используется максимальная длительность цикла, устанавливаемая по умолчанию (1000 мсек.)

- 1,2,3...**
1. Когда CIO 000000 переводится в состояние ON, первая команда WDT(094) увеличивает длительность цикла на 300 мсек. (30...10 мсек.). Таким образом, в этой точке максимальная длительность цикла становится равной 1300 мсек.
  2. Когда CIO 000001 переводится в состояние ON, вторая команда WDT(094) предпринимает попытку увеличения длительности цикла еще на 39000 мсек. Так как новая длительность цикла (40300 мсек.) превышает максимально-допустимое значение 40000 мсек., излишние 300 мсек. игнорируются. В результате вторая команда WDT(094) увеличивает максимальную длительность цикла на 38700 мсек.
  3. Когда CIO 000003 переводится в состояние ON, третья команда WDT(094) предпринимает попытку увеличения длительности цикла еще на 1000 мсек. Так как максимальная длительность цикла достигла своего предельного значения, последняя команда не выполняется.



### 3-30 Команды программирования блоков

Данный раздел описывает блочные программы и команды программирования блоков.

Команда	Мнемоническое обозначение	Функциональный код	Страница
BLOCK PROGRAM BEGIN	BPRG	096	681
BLOCK PROGRAM END	BEND	801	
BLOCK PROGRAM PAUSE	BPPS	811	684
BLOCK PROGRAM RESTART	BPRS	812	
CONDITIONAL BLOCK EXIT (NOT)	EXIT (NOT)	806	691
IF (NOT)	IF (NOT)	802	686
ELSE	ELSE	803	
IF END	IEND	804	
ONE CYCLE AND WAIT (NOT)	WAIT (NOT)	805	693
TIMER WAIT	TIMW	813	697
COUNTER WAIT	CNTW	814	700
HIGH-SPEED TIMER WAIT	TMHW	815	703
LOOP	LOOP	809	705
LOOP END	LEND (NOT)	810	

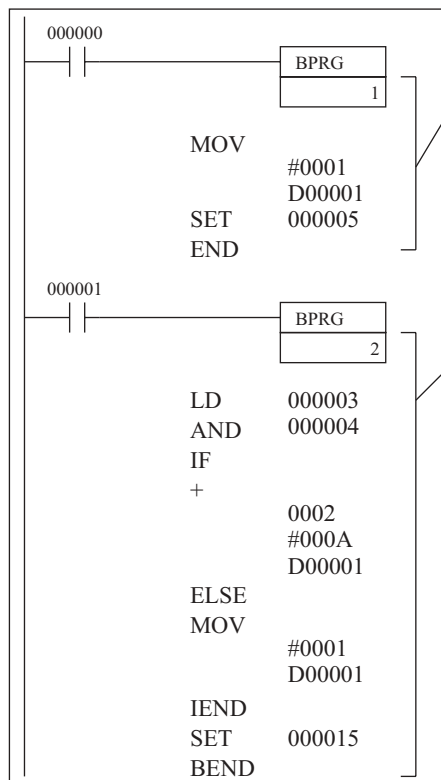
#### 3-30-1 Введение

##### Блочные программы

Пользовательская программа при использовании контроллеров CS1 может содержать до 128 блочных программ (все задачи). Выполнение каждого блока управляется простым условием выполнения. Все команды, находящиеся между BPRG(096) и BEND(801) выполняются безусловно, когда условие выполнения команды BPRG(096) переводится в состояние ON. Условие выполнения не влияет на команды программирования блоков, кроме команды BPRG(096). Это позволяет программировать в одну блочную программу все действия, запускаемые по одному условию выполнения.

Каждый из блоков релейно-контактной схемы запускается по одному условию выполнения. Все команды блока записываются в мнемонической форме. Таким образом, блочная программа является комбинацией команд релейно-контактной схемы и команд в мнемонической форме.

Блочные программы разрешают программирование действий, которые достаточно трудно запрограммировать с помощью релейно-контактной схемы, например условные ответвления или шаговые прогрессии. Следующий ниже пример показывает две блочные программы.



Область блочной программы 1.  
 Когда CIO 000000 переводится в состояние ON, выполняется блочная программа 1. Команды MOV(021) и SET выполняются безусловно, затем блочная программа завершается.

Область блочной программы 2.  
 Когда CIO 000001 переводится в состояние ON, выполняется блочная программа 2. Если CIO 000003 и CIO 000004 одновременно находятся в состоянии ON, выполняется сложение двоичных чисел (CIO 0002 + #000A D00001). Если один из этих битов находится в состоянии OFF, #0001 перемещается в D00001. После этого бит CIO 000015 устанавливается безусловно и программа завершается.

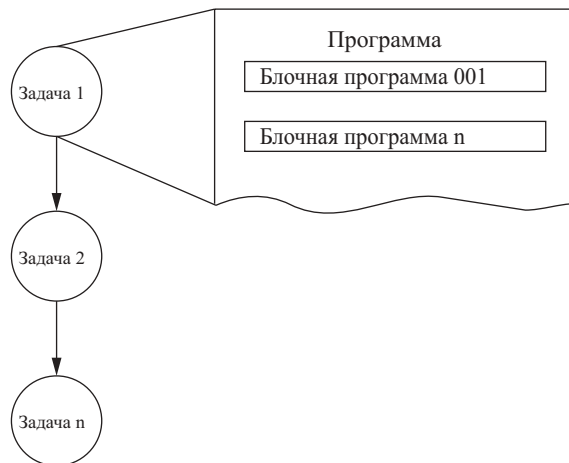
### Задачи и блочные программы

Блочные программы могут располагаться внутри задач. Задачи применяются для подразделения значительных по объему программных единиц на более мелкие, блочные программы используются в задачах для дальнейшего разделения программы на небольшие единицы, управляемые в релейно-контактной схеме простыми условиями выполнения.

Подобно задачам, блочные программы которые не должны выполняться (т.е. блоки, условия выполнения которых находятся в состоянии OFF) не требуют времени выполнения и могут использоваться для сокращения длительности цикла (что-то подобное переходам). Кроме того, подобно задачам, другие блоки могут на время приостанавливаться или запускаться из выполняемой блочной программы.

Тем не менее, между задачами и блочными программами существуют различия. Одним из различий является то, что условия ввода не применяются с блочными программами до тех пор, пока не будут преднамеренно запрограммированы с помощью команд IF(802), WAIT(805), EXIT(806), IEND(801) или с помощью прочих команд. Кроме того, существует ряд команд, которые не могут использоваться в блочных программах, например дифференцированные команды.

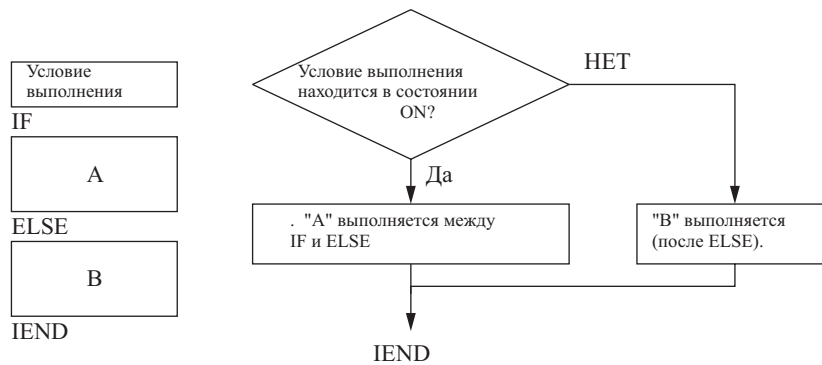
Блочные программы можно использовать как в циклических задачах, так и в задачах прерывания. Номера блочных программ от 0 до 127 могут использоваться только один раз. Не допускается применение одного номера даже в различных задачах.



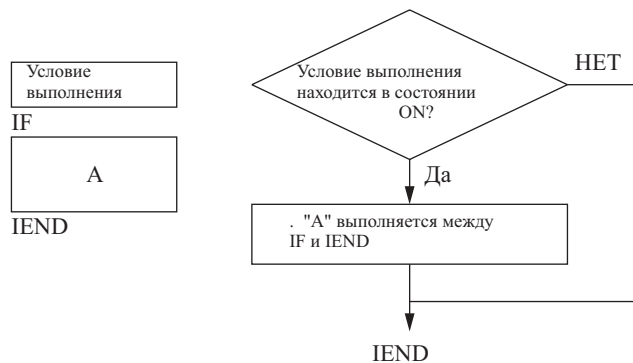
### Применение команд программирования блоков

В основном, команды IF(802), ELSE(803) и IEND(810) применяются в качестве условий выполнения внутри блочной программы (наряду с битами).

Если «А» или «В» должны выполняться, команды IF A ELSE B IEND используются, как показано ниже.



Если должно выполняться действие «А» или не одно из действий не должно выполняться, IF A IEND используются, как показано ниже.



Если выполнение команды откладывается до перевода условия выполнения или бита в состояние ON (т.е. для выполнения последовательности шагов), применяется команда WAIT(805).

Если выполнение откладывается на указываемый промежуток времени (т.е. для рассчитанного по времени выполнения последовательности шагов), применяется команда TIMW(813) или TMHW(815).

Если выполнение команды производится после достижения счетчиком заданного значения (т.е. для выполнения последовательности шагов со счетчиком), применяется команда CNTW(814).

Если выполнение должно повторяться в части блочной программы до возникновения условия, применяются команды LOOP(809) и LEND(810).

Если выполнение блочной программы должно прекращаться в середине программы в соответствии с условием выполнения, применяется команда EXIT(806).

Если другая блочная программа должна приостанавливаться или запускаться из блочной программы, используются команды BPPS(811) и BPRS(812).

### Команды, принимающие условие выполнения в блочной программе

В следующей ниже таблице приведены команды, которые в блочной программе могут принимать условие выполнения.

Тип команды	Наименование команды	Мнемоническое обозначение
Команды программирования блока	IF(NOT)	IF(802)(NOT)
	ONE CYCLE AND WAIT (NOT)	WAIT(805)(NOT)
	EXIT	EXIT(806)(NOT)
	LOOP END	LEND(810)(NOT)
Команды релейно-контактной схемы	CONDITIONAL JUMP	CJP(510)
	CONDITIONAL JUMP NOT	CJPN(511)

### Команды, имеющие ограничения к применению в блочных программах

Команды, перечисленные в следующей ниже таблице, могут использоваться только для создания условий выполнения команд IF(802), WAIT(805), EXIT(806), LEND(810), CJP(510) или CJPN(511) и не могут использоваться самостоятельно. Выполнение данных команд может привести к непредсказуемым результатам, если они применяются самостоятельно или в комбинации с другими командами.

Мнемоническое обозначение	Наименование
LD/LDNOT	LOAD/LOAD NOT
AND/AND NOT	AND/AND NOT
OR/OR NOT	OR/OR NOT
UP/DOWN	CONDITION ON/CONDITION OFF
>, <, =, >=, <=, <>, (S)(L)	Символьные команды сравнения (не являющиеся правосторонними командами).
LD TST/TST NOT	Команды загрузки теста LOAD Bit Test.
AND TST/TST NOT	Команды теста AND Bit Test.
OR TST/TST NOT	Команды теста OR Bit Test.
>\$, <\$, =\$, <=\$, >=\$, <>\$	Команды сравнения текстовой строки.

О Пример правильного применения команд

LD 000000	] Используется в качестве условия выполнения для команды IF
AND 000100	
TST D00000 #0010	
IF	

X Пример неправильного применения команд

LD 000000	] Не может использоваться в качестве условия выполнения для команды MOV(021)
AND 000100	
TST D00000 #0010	
MOV #0000 0010	

### Команды, которые не применяются в блочных программах

Команды, перечисленные в следующей ниже таблице, не могут использоваться в блочных программах.

Группа команд	Мнемоническое обозначение	Наименование	Альтернатива
Команды последовательного вывода	OUT	OUPUT	Используйте команды SET и RESET.
	OUT NOT	OUTPUT NOT	
	DIFU(013)	DIFFERENTIATE UP	Нет.
	DIFD(014)	DIFFERENTIATE DOWN	Нет.
	KEEP(011)	KEEP	Нет.
Команды последовательного управления	FOR(512) и NEXT(513)	FOR- NEXT LOOPS	Используйте команды LOOP(809) и LEND(810)(NOT)
	BREAK(514)	BREAK LOOP	
	IL(002) и ILC(003)	INTERLOCK INTERLOCK CLEAR	и Осуществляет разделение блочной программы на более мелкие блоки.
	JMP(004)0 и JME(005)0	Multiple JUMP и Multiple JUMP END	Используйте JMP(004) и JME(005) (однако переход будет производиться безусловно.)

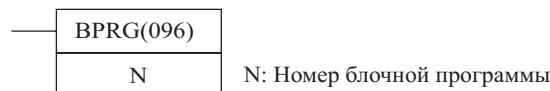
Группа команд	Мнемоническое обозначение	Наименование	Альтернатива
	END(001)	END	Используйте команду BEND(801)
Команды управления таймерами и счетчиками.	TIM	TIMER	Используйте команды TIMW(813), TMHW(815) CNTW(814). Другие команды блока выполняться не будут до истечения времени таймера или окончания счета счетчика.
	TIMH(015)	HIGH-SPEED TIMER	
	TMHH(540)	ONE MS TIMER	
	TTIM(087)	ACCOMULATIVE TIMER	
	TIML(542)	LONG TIMER	
	MTIM(543)	MULTI OUTPUT TIMER	
	CNT	COUNTER	
	CNTR(012)	REVERSIBLE COUNTER	
Команды подпрограмм	SBN(092) и RET(093)	SUBROUTINE ENTRY и SUBROUTINE RETURN	Нет.
Команды смещения	SFT(010)	SHIFT REGISTER	Используйте другие команды смещения.
Шаговые команды	STEP(???)и SNXT(???)	STEP и STEP NEXT	Используйте команду WAIT(805).
Команды управления данными	PID(190)	PID CONTROL	Нет.
Команды диагностики	FPD(269)	FAILURE DETECTION POINT	Нет.
Дифференцированные команды	Мнемоническая команда, сопровождаемая символом @.	Дифференцированные вверх (по возрастанию) команды	Нет.
	Мнемоническая команда, сопровождаемая символом %.	Дифференцированные вниз (по убыванию) команды	Нет.

### 3-30-2 Команда начала/окончания блочной программы BLOCK PROGRAM BEGIN/END: BPRG(096)/BEND(801)

#### Назначение

Данная команда определяет область программирования блока. Для каждой команды BPRG(096) должна существовать соответствующая команда BEND(801).

#### Символ релейно-контактной схемы



#### Модификации BPRG(096)

Модификации	Выполняется в каждом цикле для состояния ON	BPRG(096)
	Выполняется один раз для дифференцирования вверх	Не поддерживается
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### BEND(801)

Модификации	В блочной программе всегда выполняются.
-------------	---

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
См. примечание	Да	Да	Да

**Примечание:** В начале каждой блочной программы допускается размещение только одной команды BPRG(096).

#### Операнды

##### N: Номер блочной программы

Номер программы должен находиться между 0 и 127 (в десятичном коде).

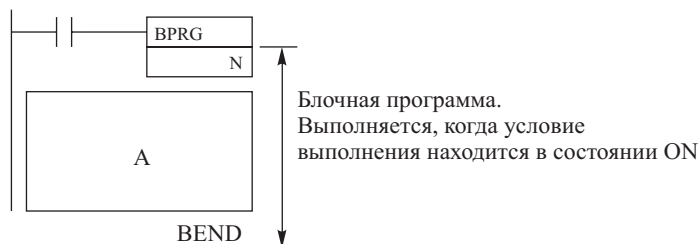


### Спецификации операндов

Область	N
Область ввода/вывода (область CIO)	—
Рабочая область	—
Область удержания бита (Holding bit area)	—
Область вспомогательных битов	—
Область таймера	—
Область счетчика	—
Область DM	—
Область EM, не содержащая банков	—
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	0...127
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	—

### Описание

По команде BPRG(096) производится выполнение блочной программы с номером, указанным в N, т.е. блока программы, начинающегося немедленно после команды BPRG(096), и заканчивающегося командой BEND(801). Все команды, находящиеся между BPRG(096) и BEND(801) выполняются с условием выполнения, находящимся в состоянии ON (т.е. безусловно).



Когда условие выполнения команды BPRG(096) находится в состоянии OFF, блочная программа не выполняется и, следовательно, для выполнения этой команды не требуется времени.

Выполнение блочной программы может останавливаться по команде BPPS(811) из другой блочной программы, даже если условие выполнения команды BPRG(096) находится в состоянии ON.

### Флаги

#### BPRG(096)

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, если команда BPRG(096) уже выполняется. Переводится в состояние ON, когда значение N выходит за пределы от 0 до 127. Переводится в состояние ON, когда один номер блочной программы используется более одного раза. OFF в других случаях.

#### BEND(801)

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, если программа не выполняется. OFF в других случаях.
Флаг равенства	=	OFF
Флаг отрицательного значения	N	OFF

**Примечание:** Метод ввода номера блока различается при использовании CX\_ программатора и Пульта программирования. При использовании CX- программатора вводите значения #0000...#007F или &0...&127, и при использовании Пульта программирования вводите значения #0000...#007F.

### Меры предосторожности

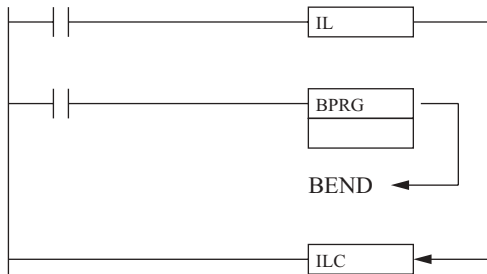
Каждый из номеров блочной программы может использоваться только один раз во всей программе пользователя.

Блочная программа не может формировать гнезда (вложения).

✗ Формирование вложений не допускается

BPRG	0
BPRG	1
BEND	
BEND	

Если блочная программа находится в заблокированной части программы, и условие выполнения для команды IL(002) находится в состоянии OFF, блочная программа не выполняется.



Команда BPRG(096) и соответствующая ей команда BEND(801) должны находиться в одной задаче.

Ошибка определяется, и флаг ошибки переводится в состояние ON в следующих случаях:

если команда BPRG(096) находится в середине блочной программы;

в блочной программе отсутствует команда BEND(801);

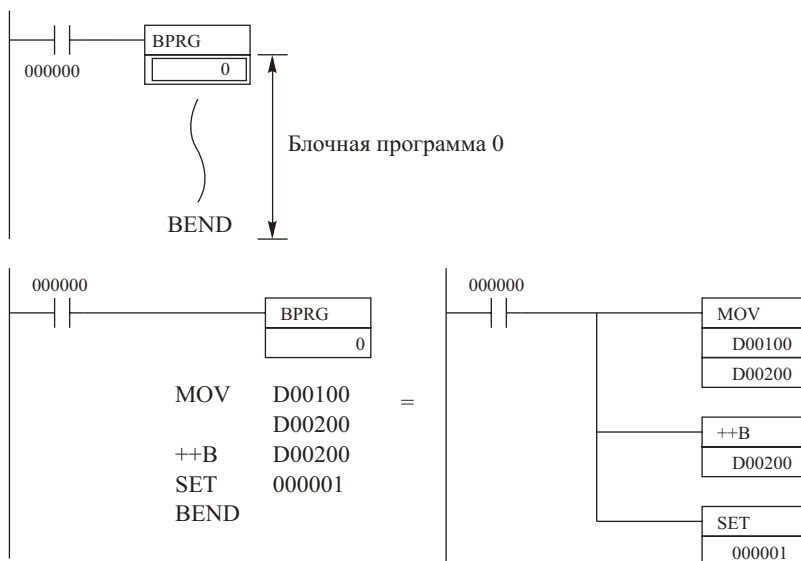
значение N выходит за пределы #0000...#007F (двоичные);

блочная программа отсутствует, или номер блочной программы используется более одного раза.

При выполнении команды BEND(801) флаг равенства и флаг отрицательного состояния переводятся в состояние OFF.

### Примеры

В следующем ниже примере, когда CIO 000000 переводится в состояние ON, производится выполнение блочной программы. Когда CIO 000000 находится в состоянии OFF, блочная программа не выполняется.



### 3-30-3 Команда приостановки/повторного запуска блочной программы BLOCK PROGRAM PAUSE/RESTART: BPPS(811)/BPRS(812)

#### Назначение

Данная команда определяет приостановку выполнения и повторный запуск блочной программы из другой блочной программы.

#### Символ релейно-контактной схемы

BPPS(811) 

N
---

 N: Номер блочной программы

BPRS(812) 

N
---

#### Модификации

Модификации	В блочной программе всегда выполняются.
-------------	---

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Примечание:** Команды BPRG(096) и BPPS(812) должны использоваться в областях программирования блоков, внутри подпрограмм и в задачах прерывания.

#### Операнды

##### N: Номер блочной программы

Номер программы должен находиться между 0 и 127 (в десятичном коде).

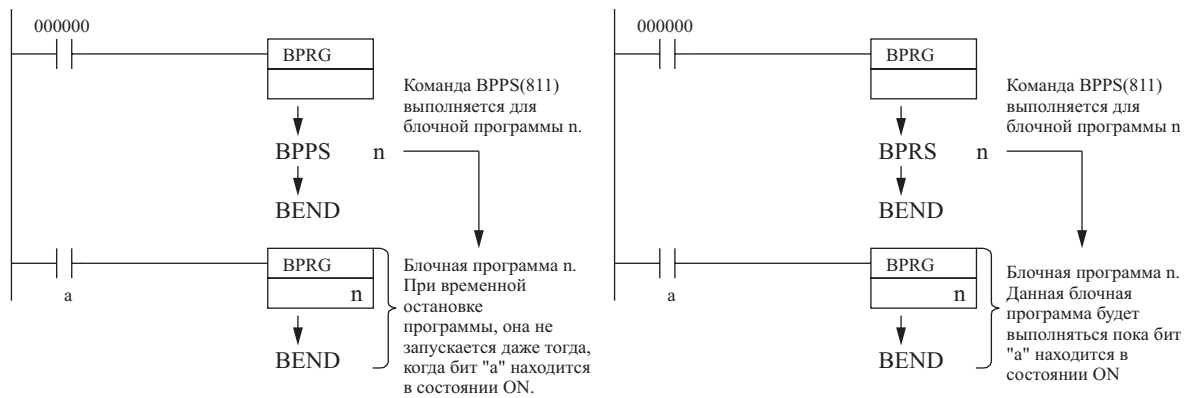
#### Спецификации операндов

Область	N
Область ввода/вывода (область CIO)	—
Рабочая область	—
Область удержания бита (Holding bit area)	—
Область вспомогательных битов	—
Область таймера	—
Область счетчика	—
Область DM	—
Область EM, не содержащая банков	—
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	0...127
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	—

#### Описание

Команда BPPS(811) используется внутри одной блочной программы для прерывания выполнения другой блочной программы, указанной номером N. Прерывание блочной программы по команде BPPS(811) производится даже тогда, когда условие выполнения команды BPRG(096) для прерываемой программы находится в состоянии ON. Повторный запуск приостановленной программы производится только после выполнения команды BPRS(812).

По команде BPRS(812) производится запуск программы, указанной в N. После первого запуска, блочная программа выполняется до тех пор, пока условие выполнения команды BPRG(096) находится в состоянии ON.



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, если команда BPPS(811) или команда BPRS(812) находится вне блока. Переводится в состояние ON, когда значение N выходит за пределы от 0 до 127. OFF в других случаях.
Флаг равенства	=	OFF
Флаг отрицательного значения.	N	OFF

**Примечание:** Метод ввода номера блока различается при использовании CX\_ программатора и Пульта программирования. При использовании CX- программатора вводите значения 0...127, и при использовании Пульта программирования вводите 0...127.

### Меры предосторожности

В следующих случаях определяется ошибка, и флаг ошибки переводится в состояние ON:  
в блочной программе отсутствует команда BPPS(811) или BPRS(812);  
значение N выходит за пределы #0000...#007F (двоичные).

Команда BPPS(811) может использоваться для временного прерывания выполнения блочной программы, которая содержит эту команду. Когда в последствии эта программа повторно запускается из другой блочной программы по команде BPRS(812), выполнение прерванной программы начинается с команды, следующей за командой BPPS(811).

Если временно прерываемая программа содержит команды TIMW(813) или TMHW(815), текущее значение времени продолжает отсчитываться в процессе прерывания программы.

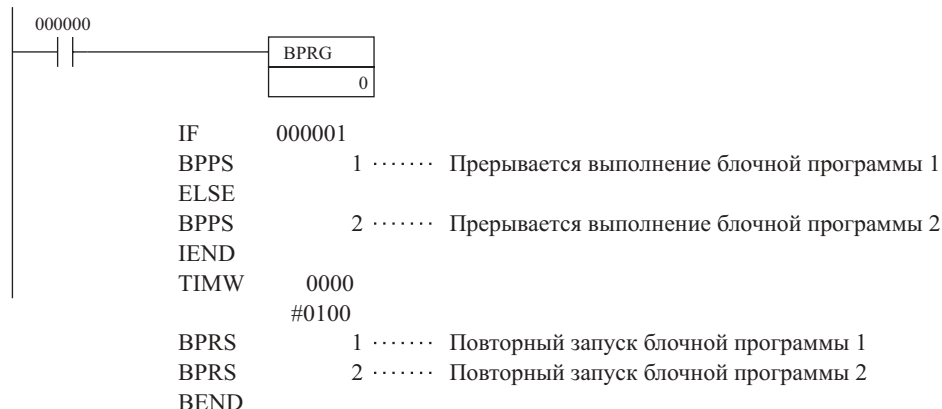
### Примеры

Следующий ниже пример показывает основной процесс прерывания программы.



**Примечание:** Если временно прерванная блочная программа появляется после команды BPPS(811), она выполняться не будет. Если блочная программа появляется перед командой BPPS(811), она будет прервана, начиная со следующего цикла.

Когда СЮ 000000 находится в состоянии ON, следующая ниже программа прерывает выполнение блочной программы 1 или блочной программы 2, в зависимости от состояния СЮ 000001. Прерванная блочная программа повторно запускается спустя 10 секунд.



Адрес	Команда	Операнды
000000	LD	000000
000001	BPRG(096)	00
000002	IF(802)	000001
000003	BPPS(811)	01
000004	ELSE(803)	
000005	BPPS(811)	02
000006	IEND(804)	
000007	TIMW(803)	0000
		# 0100
000008	BPRS(812)	1
000009	BPRS(812)	2
000010	BEND(801)	

### 3-30-4 Команды ветвления IF(802), ELSE(803) и IEND(804)

#### Назначение

Данные команды осуществляют выполнение перехода, базируясь на условии выполнения, или на состояние бита управления.

#### Символ релейно-контактной схемы

IF(802)            В            В: бит операнда

IF(802)

IF(802) NOT       В

ELSE(803)

IEND(804)

#### Модификации

Модификации	В блочной программе всегда выполняются.
-------------	---

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Примечание:** Команды IF(802), ELSE(803) и IEND(804) должны использоваться в областях программирования блоков внутри подпрограмм и в задачах прерывания.

#### Спецификации операндов

Область	В
Область ввода/вывода (область СЮ)	СЮ 000000...СЮ 614315

Область	В
Рабочая область	W00000...W51115
Область удержания бита (Holding bit area)	H00000...H51115
Область вспомогательных битов	A00000...A44715 A44800...A95915
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Флаги задач	TK0000...TK0031
Флаги условий	ER, CY, >, =, <, N, OF, UF, >=, <>, <=, ON, OFF, AER
Тактовые импульсы	0.02 сек. 0.1 сек. 0.2 сек. 1 сек. 1 мин.
Область DM	—
Область EM, не содержащая банков	—
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(++) ,(- -)IR0...,-(- -)IR15

### Описание

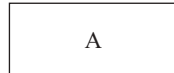
#### Работа команды IF(802) без операнда

Если бит операнда не указывается, перед командой IF(802) должно создаваться условие выполнения, начиная с команды LD. Когда условие выполнения находится в состоянии ON, команды, находящиеся между IF(802) и ELSE(803) будут выполняться, а если условие выполнения находится в состоянии OFF, будут выполняться команды, находящиеся между IF(803) и ELSE(804).

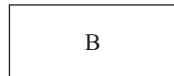
IF R (IF NOT R)



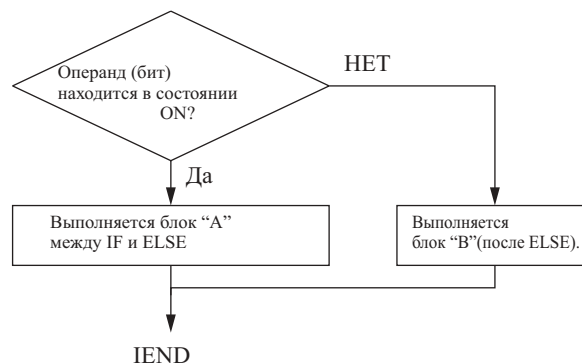
IF



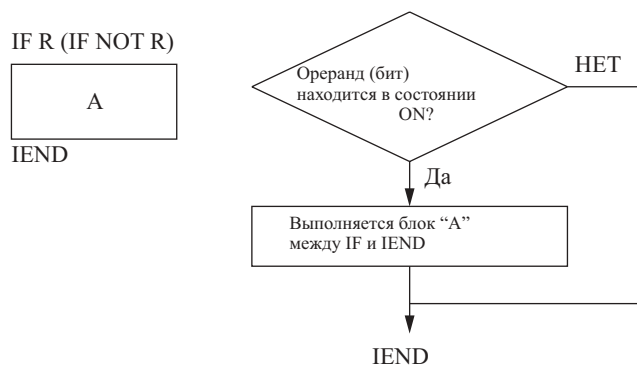
ELSE



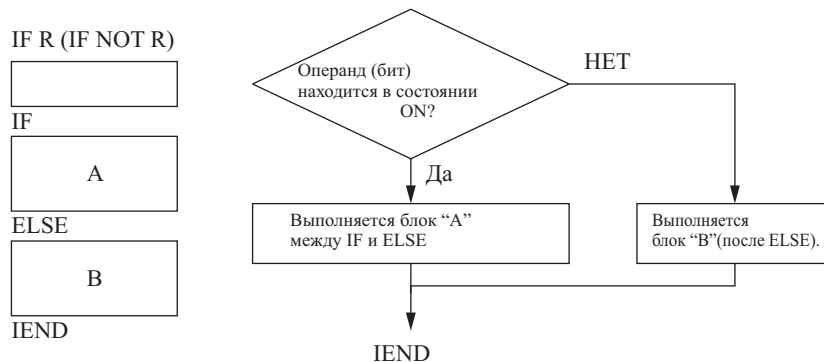
IEND



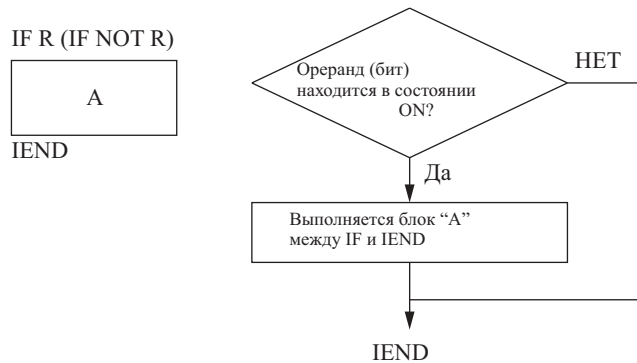
В случае, когда команда ELSE(803) опускается, а условие выполнения находится в состоянии ON, будут выполняться команды, расположенные между IF(802) и IEND(804). Если в этом случае условие выполнения находится в состоянии OFF, будут выполняться только команды, расположенные после IEND(804).

**Работа команды IF(802) с операндм**

Для команды IF(802) или IF NOT(802) может задаваться бит – операнд В. Когда этот операнд находится в состоянии ON, будут выполняться команды, расположенные между IF(802) и ELSE(803). Если этот операнд находится в состоянии OFF, будут выполняться команды, расположенные между ELSE(803) и IEND(804). Что касается IF NOT(802), если операнд находится в состоянии ON, будут выполняться команды, расположенные между IF(802) и ELSE(803), в противном случае будут выполняться команды ELSE(803) и IEND(804).



В случае, когда команда ELSE(803) опускается, а операнд находится в состоянии ON, будут выполняться команды, расположенные между IF(802) и IEND(804). Если в этом случае операнд находится в состоянии OFF, будут выполняться только команды, расположенные после IEND(804). Аналогичное происходит для противоположного состояния операнда в случае использования команды IF NOT

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, если команды выполнения перехода находятся вне блочной программы. Переводится в состояние ON, когда создается более 254 ответвлений (гнезд). OFF в других случаях.
Флаг равенства	=	OFF
Флаг отрицательного значения.	N	OFF

**Меры предосторожности**

Команды блочной программы обычно выполняются без условий. Тем не менее, для выполнения команд, базируясь на условиях выполнения или на состояние бита-операнда, может использоваться переход (ответвление).

Используйте команды IF A ELSE B IEND для выполнения перехода между A и B.

Используйте команды IF A IEND для выполнения перехода между A и прекращением работы.

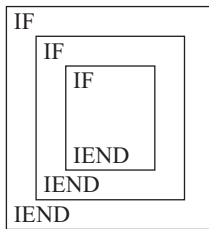
Допускается выполнение до 253 ответвлений.

В случае, когда команды выполнения перехода находятся вне блочной программы, или выполняется более 253 переходов, определяется ошибка и флаг ошибки переводится в состояние ON.

При выполнении команд IF(802), ELSE(803) или IEND(804) флаг равенства и флаг отрицательного значения переводятся в состояние OFF.

**Выполнение переходов**

В ответвление самого высокого уровня может вкладываться до 253 ответвлений (переходов).

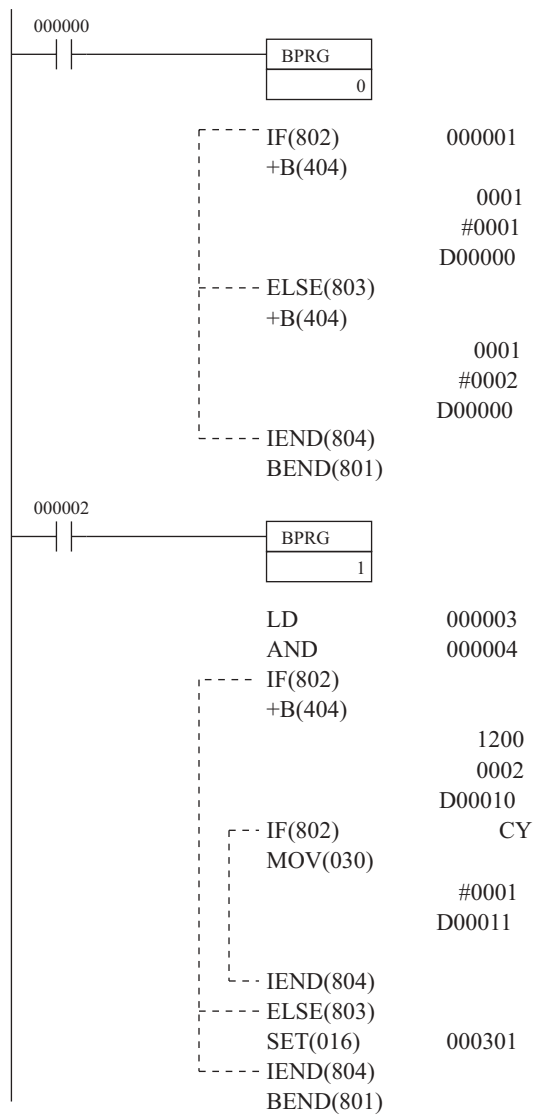
**Примеры**

В следующем ниже примере показаны две различные блочные программы, управляемые СЮ 000000 и СЮ 000002.

Первый блок выполняет одно из двух суммирований, в зависимости от состояния СЮ 000001. Этот блок выполняется, когда СЮ 000000 находится в состоянии ON. Когда СЮ 000001 находится в состоянии ON, к содержанию СЮ 0001 прибавляется 0001. Если СЮ 000001 находится в состоянии OFF, к содержанию СЮ 0001 прибавляется 0002. В обоих случаях результат заносится в D00000.

Второй блок выполняется, когда СЮ 000002 находится в состоянии ON и показывает выполнение переходов на двух уровнях. Когда СЮ 000003 и СЮ 000004 находятся в состоянии ON, содержание СЮ 1200 и СЮ 0002 суммируются, результат заносится в D00010, а затем 0001 перемещается в D00011, базируясь на состоянии флага переноса CY. Если СЮ 000003 и СЮ 000004 одновременно находятся в состоянии OFF, операция суммирования пропускается, и СЮ 000301 переводится в состояние ON.





Адрес	Команда	Операнды
000000	LD	000000
000001	BPRG(096)	00
000002	IF(802)	000001
000003	+B(404)	0001
		# 0001
		D 00000
000004	ELSE(803)	
000005	+B(404)	0001
		# 0002
		D 00000
000006	IEND(804)	
000007	BEND(801)	
000008	LD	000002
000009	BPRG(096)	1
000010	LD	000003
000011	AND	000004
000012	IF(802)	

Адрес	Команда	Операнды
000013	+B(404)	1200
		0002
		D 00010
000014	IF(802)	A 500004
000015	MOV(021)	# 0001
		D 00011
000016	IEND(804)	
000017	ELSE(803)	
000018	SET(016)	000301
000019	IEND(804)	
000020	BEND(801)	

### 3-30-5 Команда выхода из условного блока CONDITIONAL BLOCK EXIT(NOT): EXIT(NOT)(806)

#### Назначение

Осуществляет выход из блочной программы (т.е. не выполняет никакие команды блочной программы, кроме BEND(801)) в зависимости от состояния операнда (бита) или от условия выполнения. По команде EXIT(806) без операнда, производится выход из программы, если условие выполнения находится в состоянии ON. По команде EXIT(806) с операндом, производится выход из программы, если операнд находится в состоянии ON. Команда EXIT NOT(806) должна содержать бит – операнд, причем выход из программы осуществляется, когда операнд находится в состоянии OFF.

#### Символ релейно-контактной схемы

EXIT(806)

EXIT(806)            B                    B: бит операнда

EXIT NOT(806)    B

#### Модификации

Модификации	В блочной программе всегда выполняются.	EXIT(806), EXIT(806) B, EXIT NOT (806) B
-------------	---	--

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Примечание:** Команды EXIT(806) и EXIT NOT (806) должны использоваться в областях программирования блоков внутри подпрограмм и в задачах прерывания.

#### Спецификации операндов

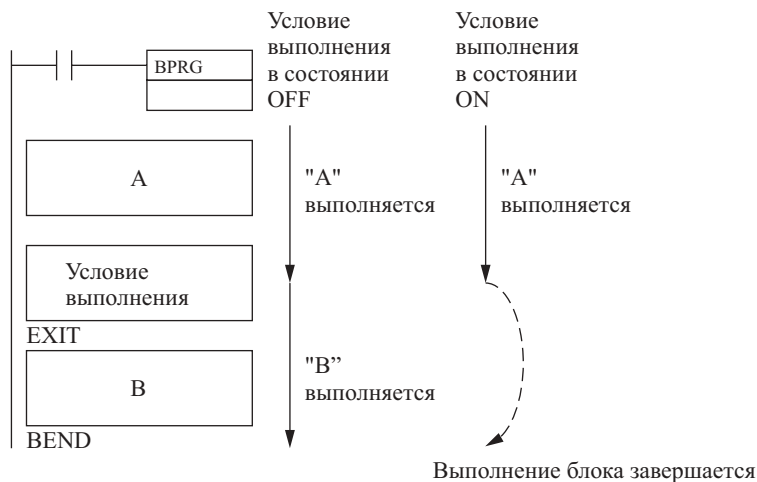
Область	B
Область ввода/вывода (область CIO)	CIO 000000...CIO 614315
Рабочая область	W00000...W51115
Область удержания бита (Holding bit area)	H00000...H51115
Область вспомогательных битов	A00000...A44715 A44800...A95915
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Флаги задач	TK0000...TK0031
Флаги условий	ER, CY, >, =, <, N, OF, UF, >=, <=, <=, ON, OFF, AER
Тактовые импульсы	0.02 сек. 0.1 сек. 0.2 сек. 1 сек. 1 мин.
Область DM	–
Область EM, не содержащая банков	–
Область EM, содержащая банки	–
Адреса DM/EM памяти при косвенном указании в двоичном коде	–
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–
Константы	–
Регистры данных	–
Индексные регистры	–

Область	В
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048 +2047,IR0...-2048...+2047, IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-(-)IR15

### Описание

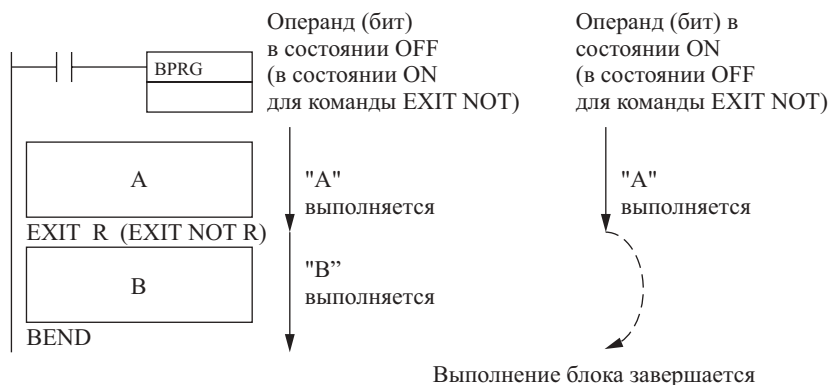
#### Работа без операнда

Команда EXIT(806) может выполняться без указания операнда. Если операнд не указывается, для ее выполнения должно создаваться условие выполнения, начиная с команды LD. Когда условие выполнения находится в состоянии OFF, остающаяся часть программы выполняется обычным путем. Когда условие выполнения находится в состоянии ON, остающаяся часть программы до команды BEND(801) не выполняется.



#### Работа с операндом

Когда операнд В команды EXIT(806) находится в состоянии OFF, остающаяся часть программы выполняется обычным путем. Когда операнд находится в состоянии ON, остающаяся часть программы до команды BEND(801) не выполняется. Что касается команды EXIT NOT(806), остающаяся часть программы выполняется, когда операнд находится в состоянии ON, и пропускается, когда операнд находится в состоянии OFF.



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, если команда EXIT(806) или EXIT NOT(806) находится вне блочной программы. OFF в других случаях.
Флаг равенства	=	OFF
Флаг отрицательного значения.	N	OFF

**Меры предосторожности**

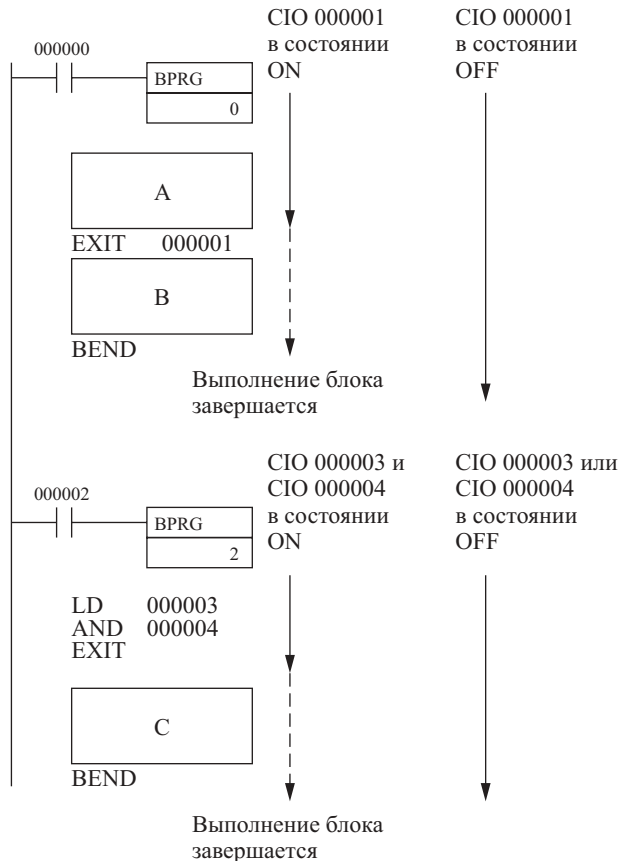
Если команда EXIT(806) или EXIT NOT(806) находится вне блочной программы, определяется ошибка и флаг ошибки переводится в состояние ON.

**Примеры**

Когда CIO 000000 находится в состоянии OFF, блочная программа выполняется. Если CIO 000001 находится в состоянии ON, выполняется «А», а «В» пропускается. Управление программой переходит к команде BEND(801).

Раздел «В» программы будет обходиться, пока CIO 000001 не будет переведен в состояние OFF.

Несмотря на то, что использование команды EXIT(NOT)(806) подобно программированию IF-IEND, в случае применения команды EXIT(NOT)(806) время выполнения обычно меньше, так как команды от EXIT(NOT)(806) до конца блочной программы не выполняются.

**3-30-6 Команда прерывания выполнения программы на один цикл ONE CYCLE AND WAIT(NOT): WAIT(805)/WAIT(805)NOT****Назначение**

По команде WAIT(805)/WAIT(805)NOT осуществляется прерывание выполнения остающейся части программы до перевода условия выполнения в состояние ON, или изменения состояние операнда (бита) в состояние OFF или ON.

**Символ релейно-контактной схемы**

WAIT(805)

WAIT(805)      В                      В: бит операнда

WAIT NOT(805) В

**Модификации**

Модификации	В блочной программе всегда выполняются.
-------------	---

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Примечание:** Команды *WAIT(805)*, и *WAIT(805) NOT* должны использоваться в областях программирования блоков внутри подпрограмм и в задачах прерывания.

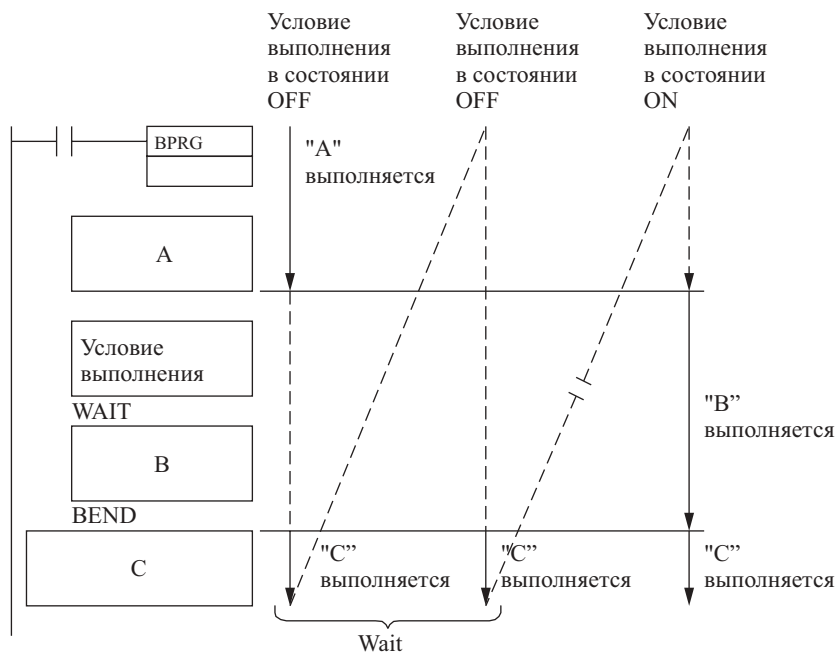
#### Спецификации операндов

Область	B
Область ввода/вывода (область CIO)	CIO 000000...CIO 614315
Рабочая область	W00000...W51115
Область удержания бита (Holding bit area)	H00000...H51115
Область вспомогательных битов	A00000...A44715 A44800...A95915
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Флаги задач	TK0000...TK0031
Флаги условий	ER, CY, >, =, <, N, OF, UF, >=, <=, ON, OFF, AER
Тактовые импульсы	0.02 сек. 0.1 сек. 0.2 сек. 1 сек. 1 мин.
Область DM	—
Область EM, не содержащая банков	—
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15

#### Описание

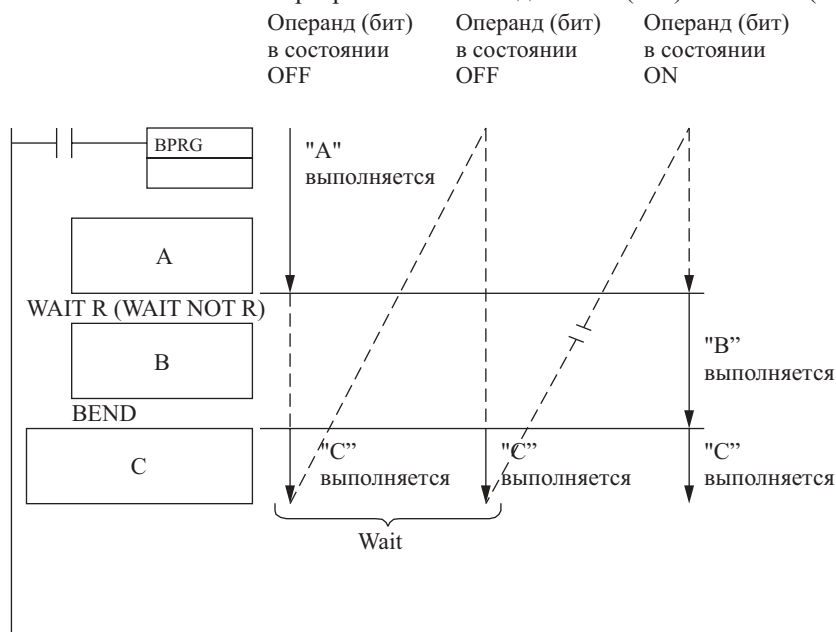
##### Работа без операнда

Если операнд не указывается, для выполнения команды перед командой *WAIT(805)*, *WAIT(805) NOT* должно создаваться условие выполнения, начиная с команды *LD*. Когда условие выполнения находится в состоянии *ON*, остающаяся часть программы от команды *WAIT(805)* до конца программы не выполняется. В следующем цикле не выполняется ни одна из блочных программ, кроме выполнения команды *WAIT(805)* при возникновении условия выполнения. Когда условие выполнения переводится в состояние *ON*, осуществляется выполнение части программы от команды *WAIT(805)* до конца программы.



#### Работа с операндом

Для команд WAIT(805) и WAIT(805) NOT может указываться операнд-бит В. Когда операнд В команды WAIT(805) находится в состоянии OFF (или в состоянии ON для команды WAIT(805)NOT), остающаяся часть программы не выполняется. В следующем цикле не выполняется ни одна из блочных программ, кроме выполнения команды WAIT(805) или WAIT(805)NOT при возникновении условия выполнения. Когда условие выполнения переводится в состояние ON (OFF для команды WAIT(805)NOT), осуществляется выполнение всей части программы от команды WAIT(805) или WAIT(805)NOT до конца программы.



#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, если команда WAIT(805) или WAIT(805) NOT находится вне блочной программы. OFF в других случаях.
Флаг равенства	=	OFF
Флаг отрицательного значения.	N	OFF

**Меры предосторожности**

Команды WAIT(805) и WAIT(805) NOT могут применяться для выполнения шагов внутри блочной программы.

Если команда WAIT(805) или WAIT(805) находится вне блочной программы, определяется ошибка и флаг ошибки переводится в состояние ON.

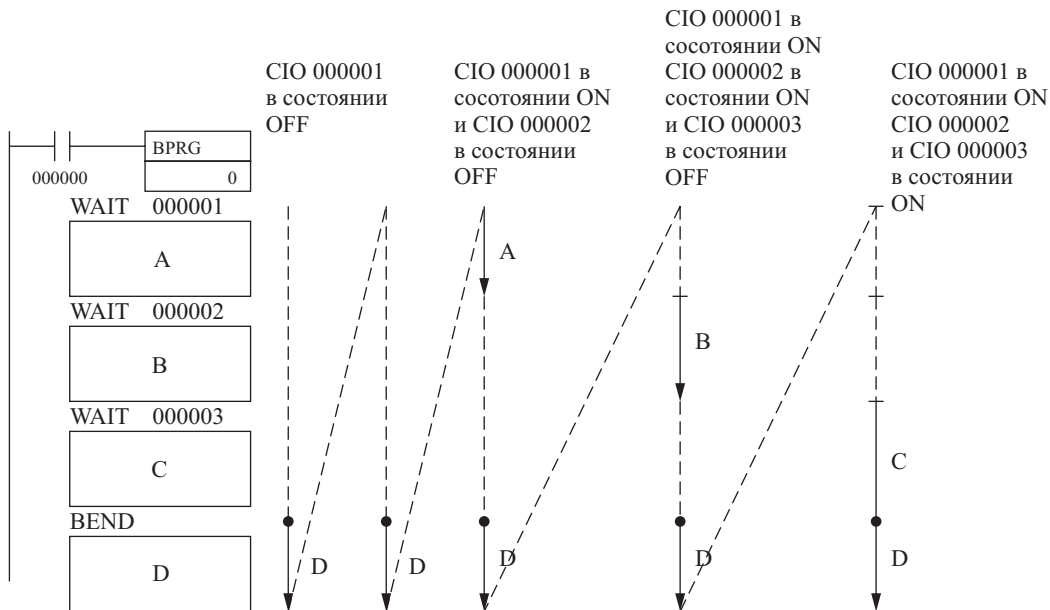
При выполнении команды WAIT(805) и WAIT(805) флаг равенства и флаг отрицательного значения переводятся в состояние OFF.

**Примечание:** Программные адреса команды WAIT с указанными операндами, а также программные адреса первой команды, создающей условия выполнения для команды WAIT без указания операндов, записываются в памяти для разрешения последующего выполнения, базируясь на условии выполнения/состоянии операнда (бита). Тем не менее, если оперативное редактирование выполняется с помощью Периферийного устройства, состояние ожидания (WAIT) будет сброшено, и блочная программа будет повторно выполняться с самого начала.

**Примеры**

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, выполняется блочная программа 00. Выполнение будет производиться следующим образом:

- Примечание:**
1. Если CIO 000001 находится в состоянии OFF, не выполняется ни одна блочная программа до тех пор, пока CIO 000001 не будет переведен в состояние ON. Когда CIO 000001 переводится в состояние ON, выполняется блок «А».
  2. Если CIO 000002 после выполнения «А» находится в состоянии OFF, оставшаяся часть блочной программы не выполняется до тех пор, пока CIO 000002 не будет переведен в состояние ON. Когда CIO 000002 переводится в состояние ON, выполняется «В».
  3. Если CIO 000003 после выполнения «В» находится в состоянии OFF, оставшаяся часть блочной программы не выполняется до тех пор, пока CIO 000003 не будет переведен в состояние ON. Когда CIO 000003 переводится в состояние ON, выполняется «С» и процесс повторяется.



В следующей ниже таблице представлены соотношения между состоянием операнда (бита) и выполнением блочной программы.

Биты операнда	Выполнение программы				
CIO 000001	CIO 000002	CIO 000003	Первый цикл, CIO 000000 в состоянии ON.	Следующий цикл	Последующие циклы
OFF	Любое состояние	Любое состояние	Ничего не выполняется.	Ничего не выполняется; ожидание CIO 000001.	Когда CIO 000001 переводится в состояние ON, выполняется «А» и контролируется состояние CIO 000002.





Область	N	SV
Адреса DM/EM памяти при косвенном указании в двоичном коде	—	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)
Константы	—	#0000...#9999 (двоично-десятичные)
Регистры данных	—	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...IR15(++) ,(- -)IR0...(- -)IR15	

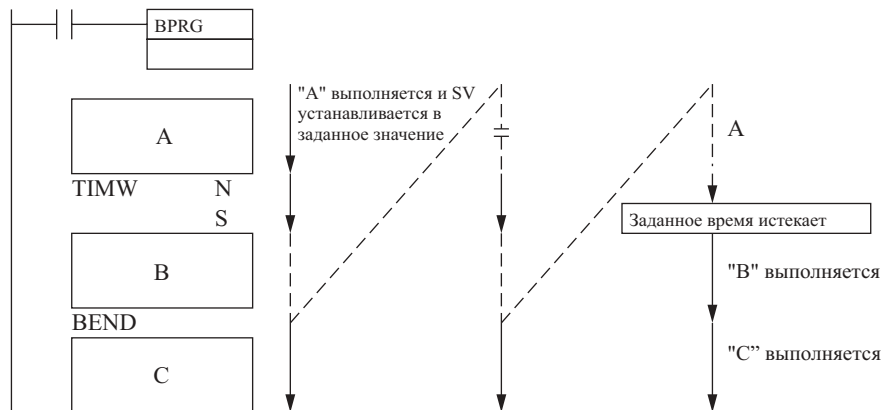
### Описание

По команде TIMW(813) создается таймер с обратным отсчетом (установка заданного значения в единицах, равных 100 мсек.), при этом между выполнением команды, предшествующей команде TIMW, и выполнением команды, следующей за командой TIMW, устанавливается задержка. Команда TIMW(813) может создавать задержку от 0 до 999.9 сек. с точностью от 0 до 0.01 сек.

Первая часть блочной программы выполняется при первом вводе программы. При достижении программой команды TIMW(813) флаг завершения переводится в состояние OFF, таймер устанавливается в заданное значение, и выполнение оставшейся части программы откладывается до завершения таймером отсчета.

В процессе выполнения таймером обратного отсчета, будет выполняться только команда TIMW(813) для обновления значения таймера. После истечения заданного времени, флаг завершения переводится в состояние ON, и начинается выполнения оставшейся части программы. После выполнения всего блока, процесс повторяется.

Команду TIMW(813) можно представить, как команду WAIT с таймером в качестве условия выполнения, поэтому эта команда может применяться для хронометрированного выполнения последовательных шагов.



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, если команда TIMW(813) находится вне блочной программы. Переводится в состояние ON, когда для N применяется косвенная адресация с использованием индексных регистров, и указываемый адрес не предназначен для текущего значения таймера. Переводится в состояние ON, когда заданное значение выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	OFF
Флаг отрицательного значения.	N	OFF

### Меры предосторожности

Остающаяся часть блочной программы, следующая за командой TIMW(813), будет выполняться в случае, когда флаг завершения таймера принудительно устанавливается.

Когда флаг завершения принудительно переустанавливается, в блочной программе выполняется только команда TIMW(813) до тех пор, пока принудительно установленное состояние не будет сброшено.

Текущее состояние таймеров с номерами от 0000 до 2047 обновляется, даже когда таймер находится в состоянии ожидания. Когда таймеры с номерами от 2048 до 4095, находятся в состоянии ожидания, текущее состояние таймеров удерживается.

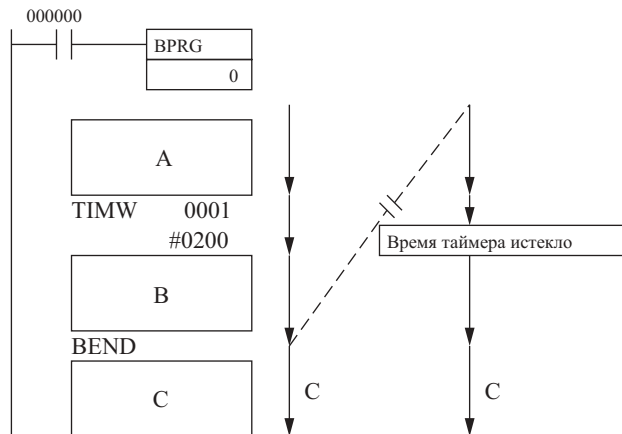
Номера таймеров используются также другими командами управления таймерами. Однако результат выполнения операций может быть непредсказуемым, если один и тот же номер используется более чем одной командой. Используйте каждый из номеров таймеров только один раз. Единственным путем использования одного номера более чем один раз, является обеспечение независимого использования, когда в одно время работает только один таймер. В случае одновременного использования одного и того же номера в различных командах, при проверке программы определяется ошибка.

Если для N применяется косвенная адресация с использованием индексного регистра и указываемый адрес не предназначен для текущего значения таймера, или когда заданное значение выражено не в двоично-десятичном коде, определяется ошибка и флаг ошибки переводится в состояние ON.

При выполнении команды TIMW(813) флаг равенства и флаг отрицательного значения переводятся в состояние OFF.

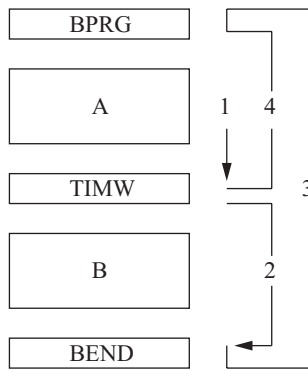
### Примеры

В следующем ниже примере, «В» выполняется спустя 20 секунд после «А» в любой момент, когда СЮ 000000 находится в состоянии ON.



Адрес	Команда	Операнд
000200	LD	000000
000201	BPRG	0
...	A	...
000210	TIMW	0001
...	B	...
000220	BEND	-

Выполнение программы производится от 2 к 3 к 4 и опять к 2 в течение 20 сек. перед выполнением «В», как показано на следующем ниже рисунке.



### 3-30-8 Команда ожидания счетчика COUNTER WAIT: CNTW(814)

#### Назначение

По команде CNTW(814) осуществляется задержка выполнения остающейся части блочной программы до достижения счетчиком заданного значения. Выполнение программы будет продолжаться с команды, следующей за командой CNTW(814), когда счетчик закончит отсчет.

#### Символ релейно-контактной схемы

CNTW(814)	N	N: номер счетчика
	SV	SV: заданное значение
	I	I: счетный вход

#### Модификации

Модификации	В блочной программе всегда выполняются.
-------------	---

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Примечание:** Команда CNTW(814) должна применяться в области программирования блоков в подпрограммах и задачах прерывания.

#### Операнды

##### N: Номер счетчика

Номер счетчика должен иметь значения от 0000 до 4095.

#### Спецификации операндов

Область	N	SV	I
Область ввода/вывода (область CIO)	–	CIO 0000...CIO 6143	CIO 000000...CIO 614315
Рабочая область	–	W000...W511	W00000...W51115
Область удержания бита (Holding bit area)	–	H000...H511	H00000...H51115
Область вспомогательных битов	–	A000...A447 A448...A959	A00000...A44715 A44800...A95915
Область таймера	–	T0000...T4095	T0000...T4095
Область счетчика	C0000...C4095	C0000...C4095	C0000...C4095
Флаги задач	–	TK0000...TK0031	
Флаги условий	–	ER, CY, >, =, <, N, OF, UF, >=, <>, <=, ON, OFF, AER	
Тактовые импульсы	–	0.02 сек. 0.1 сек. 0.2 сек. 1 сек. 1 мин.	
Область DM	–	D00000...D32767	–
Область EM, не содержащая банков	–	E00000...E32767	–
Область EM, содержащая банки	–	En_00000...En_32767 (n=0...C)	–
Адреса DM/EM памяти при косвенном указании в двоичном коде	–	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 7 (n=0...C)	–

Область	N	SV	I
intbl Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	–	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)	–
Константы	–	#0000...#FFFF (двоичные)	–
Регистры данных	–		
Индексные регистры	–		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,-(-)IR0...,-(-)IR15		

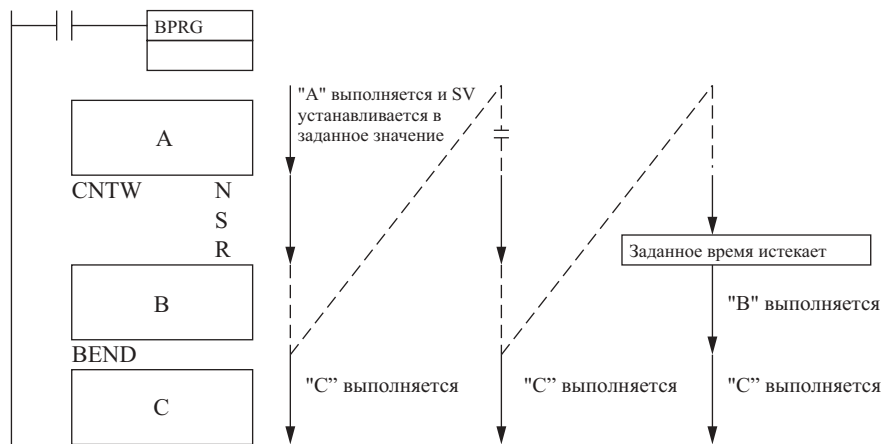
### Описание

По команде CNTW(814) создается счетчик с обратным отсчетом, который откладывает выполнение команд, следующих за счетчиком, до завершения отсчета. Заданное значение указывается в двоично-десятичном коде от 0000 до 9999.

Первая часть блочной программы выполняется при первом вводе программы. При достижении программой команды CNTW(814) флаг завершения переустанавливается в значение 0, счетчик устанавливается в заданное значение, и выполнение оставшейся части программы откладывается до завершения счетчиком отсчета. Счетчик подсчитывает импульсы (дифференцирование вверх) на вводе I.

В процессе выполнения счетчиком обратного отсчета, будет выполняться только команда CNTW(814) для обновления значения счетчика. После завершения отсчета, флаг завершения переводится в состояние ON, и начинается выполнения оставшейся части программы. После выполнения всей блочной программы, процесс повторяется.

Команду CNTW(814) можно представить, как команду WAIT со счетчиком в качестве условия выполнения, поэтому эта команда может применяться для хронометрированного выполнения последовательных шагов.



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, если команда CNTW (814) находится вне блочной программы. Переводится в состояние ON, когда для N применяется косвенная адресация с использованием индексных регистров, и указываемый адрес не предназначен для текущего значения счетчика. Переводится в состояние ON, когда заданное значение выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	OFF
Флаг отрицательного значения.	N	OFF

### Меры предосторожности

Остающаяся часть блочной программы, следующая за командой CNTW(814), будет выполняться в случае, когда флаг завершения счетчика принудительно устанавливается.

Когда флаг завершения принудительно переустанавливается, в блочной программе выполняется только команда CNTW(814) до тех пор, пока принудительно установленное состояние не будет сброшено.

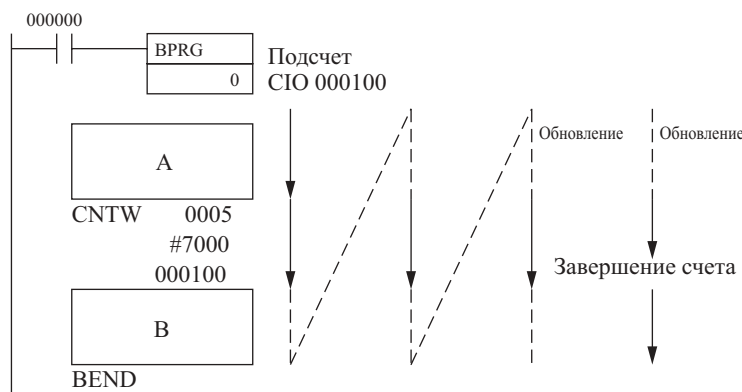
Номера счетчиков используются также другими командами управления. Однако результат выполнения операций может быть непредсказуемым, если один и тот же номер используется более чем одной командой. Используйте каждый из номеров счетчиков только один раз. Единственным путем использования одного номера более чем один раз, является обеспечение независимого использования, когда в одно время работает только один счетчик. В случае одновременного использования одного и того же номера в различных командах, при проверке программы определяется ошибка.

Если для N применяется косвенная адресация с использованием индексного регистра и указываемый адрес не предназначен для текущего значения счетчика, или когда заданное значение выражено не в двоично-десятичном коде, определяется ошибка и флаг ошибки переводится в состояние ON.

При выполнении команды CNTW(814) флаг равенства и флаг отрицательного значения переводятся в состояние OFF.

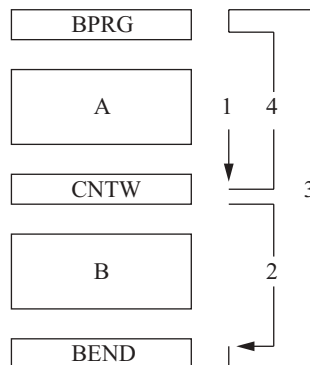
### Примеры

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, выполняется «А», а выполнение остающейся части программы «В» откладывается до достижения счетчиком CIO000100 значения 7000.



Адрес	Команда	Операнд
000200	LD	000000
000201	BPRG	0
...	A	...
000210	CNTW	0005
		# 7000
...	B	...
000220	BEND	-

Выполнение программы производится от 2 к 3 к 4 и опять к 2 в течение отсчета до значения 7000 перед выполнением «В», как показано на следующем ниже рисунке.



## 3-30-9 Команда ожидания высокоскоростного таймера HIGH SPEED TIMER WAIT: TMHW(815)

**Назначение**

По команде TMHW(815) осуществляется задержка выполнения остающейся части блочной программы до истечения заданного времени. Выполнение программы будет продолжаться с команды, следующей за командой TMHW(815), когда таймер закончит отсчет времени.

**Символ релейно-контактной схемы**

TMHW(815)      N              N: номер таймера  
                       SV              SV: заданное значение

**Модификации**

Модификации	В блочной программе всегда выполняются.
-------------	---

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Примечание:** Таймер TMHW(815) должен использоваться в области программирования блока в подпрограммах.

**Операнды****N: Номер таймера**

Номер таймера должен иметь значения от 0000 до 4095.

**Спецификации операндов**

Область	N	SV
Область ввода/вывода (область CIO)	—	CIO 0000...CIO 6143
Рабочая область	—	W000...W511
Область удержания бита (Holding bit area)	—	H000...H511
Область вспомогательных битов	—	A000...A447 A448...A959
Область таймера	T0000...T4095	T0000...T4095
Область счетчика	—	C0000...C4095
Область DM	—	D00000...D32767
Область EM, не содержащая банков	—	E00000...E32767
Область EM, содержащая банки	—	En_00000...En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	—	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)
Константы	—	#0000...#FFFF (двоичные)
Регистры данных	—	DR0...DR15
Индексные регистры	—	—
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15(++) ,-(-)IR0...,-(-)IR15	

**Описание**

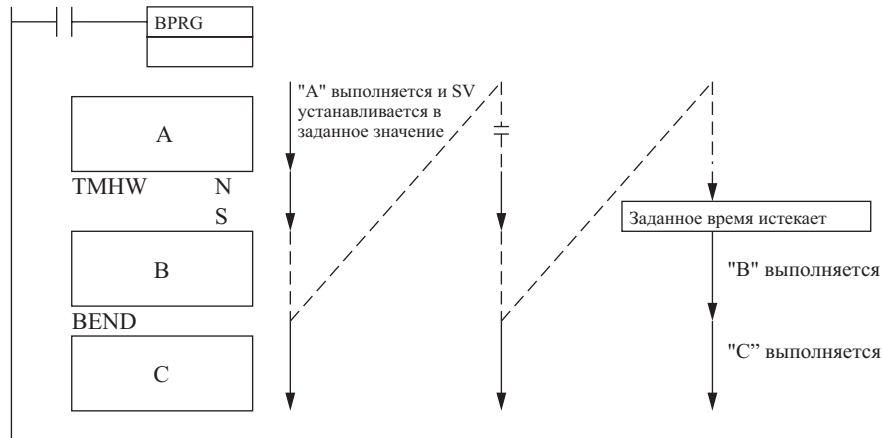
По команде TMHW(815) создается таймер с обратным отсчетом (установка заданного значения в единицах, равных 10 мсек.), при этом между выполнением команды, предшествующей команде TMHW, и выполнением команды, следующей за командой TMHW, устанавливается задержка. Команда TMHW(815) может создавать задержку от 0 до 99.99 сек. с точностью от 0 до 0.01 сек.

Первая часть блочной программы выполняется при первом вводе программы. При достижении программой команды TMHW(815) флаг завершения переводится в состояние OFF, таймер устанавливается в заданное значение, и выполнение оставшейся части программы откладывается до завершения таймером отсчета.

В процессе выполнения таймером обратного отсчета, будет выполняться только команда TMHW(815) для обновления значения таймера. После истечения заданного времени, флаг завершения переводится в со-

стояние ON, и начинается выполнения оставшейся части программы. После выполнения всего блока, процесс повторяется.

Команду TMHW(815) можно представить в виде команды WAIT с таймером в качестве условия выполнения, поэтому эта команда может применяться для хронометрированного выполнения последовательных шагов.



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, если команда TMHW(815) находится вне блочной программы. Переводится в состояние ON, когда для N применяется косвенная адресация с использованием индексных регистров, и указываемый адрес не предназначен для текущего значения таймера. Переводится в состояние ON, когда заданное значение выражено не в двоично-десятичном коде. OFF в других случаях.
Флаг равенства	=	OFF
Флаг отрицательного значения.	N	OFF

### Меры предосторожности

Остающаяся часть блочной программы, следующая за командой TMHW(815), будет выполняться в случае, когда флаг завершения таймера принудительно устанавливается.

Когда флаг завершения принудительно переустанавливается, в блочной программе выполняется только команда TMHW(815) до тех пор, пока принудительно установленное состояние не сбрасывается.

Текущее состояние таймеров с номерами от 0000 до 2047 обновляется, даже когда таймер находится в состоянии ожидания. Когда таймеры с номерами от 2048 до 4095, находятся в состоянии ожидания, текущее состояние таймеров удерживается.

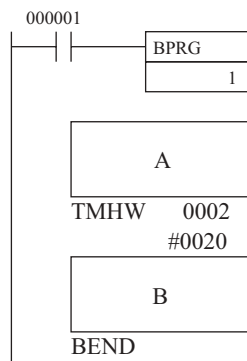
Номера таймеров используются также другими командами управления таймерами. Однако результат выполнения операций может быть непредсказуемым, если один и тот же номер используется более чем одной командой. Используйте каждый из номеров таймеров только один раз. Единственным путем использования одного номера более чем один раз, является обеспечение независимого использования, когда в одно время работает только один таймер. В случае одновременного использования одного и того же номера в различных командах, при проверке программы определяется ошибка.

Если для N применяется косвенная адресация с использованием индексного регистра и указываемый адрес не предназначен для текущего значения таймера, или когда заданное значение выражено не в двоично-десятичном коде, определяется ошибка и флаг ошибки переводится в состояние ON.

При выполнении команды TIMW(813) флаг равенства и флаг отрицательного значения переводятся в состояние OFF.

### Примеры

В следующем ниже примере, «B» выполняется спустя 20 секунд после «A» в любой момент, когда CIO 000000 находится в состоянии ON.



Адрес	Команда	Операнд
000221	LD	000001
000222	BPRG	1
...	A	...
000250	TMHW	0002
		# 0020
...	B	...
000281	BEND	–

### 3-30-10 Команды управления циклом LOOP(809)/LEND(810)/ LEND(810) NOT

#### Назначение

По команде управления циклом создается цикл, который постоянно повторяется до тех пор, пока условие выполнения не изменяет своего состояния, или до тех пор, пока условие выполнения не переведется в состояние ON.

#### Символ релейно-контактной схемы

LOOP(809)

LEND(810)

LEND(810) B B: бит операнда

LEND NOT (810) B

#### Модификации

Модификации	В блочной программе всегда выполняются.
-------------	---

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Примечание:** Команды LOOP(809)/LEND(810)/LEND(810) NOT должны использоваться в области программирования блока в подпрограммах и задачах прерывания.

#### Спецификации операндов

Область	B
Область ввода/вывода (область CIO)	CIO 000000...CIO 614315
Рабочая область	W00000...W51115
Область удержания бита (Holding bit area)	H00000...H51115
Область вспомогательных битов	A00000...A44715 A44800...A95915
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Флаги задач	TK0000...TK0031
Флаги условий	ER, CY, >, =, <, N, OF, UF, >=, <>, <=, ON, OFF, AER
Тактовые импульсы	0.02 сек. 0.1 сек. 0.2 сек. 1 сек. 1 мин.
Область DM	–
Область EM, не содержащая банков	–
Область EM, содержащая банки	–



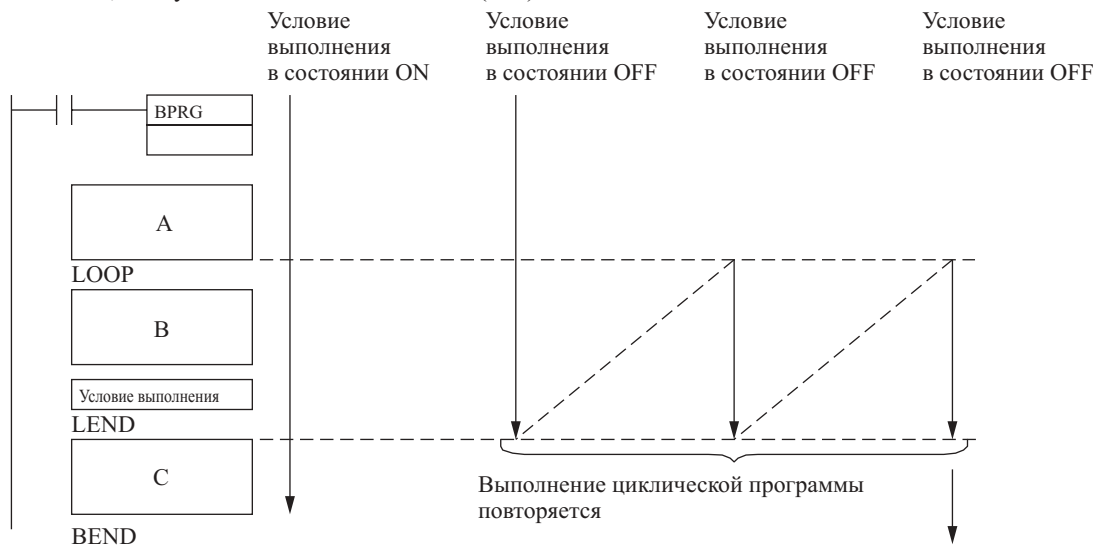
Область	В
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,(- -)IR0...,-(- -)IR15

### Описание

Команда LOOP(809) указывает начало циклической программы. Команда LEND(810) или LEND (810) NOT указывает конец циклической программы. При достижении команд LEND(810) или LEND (810) NOT выполнение программы переходит к предшествующей команде LOOP(809). Это происходит до тех пор, пока операнд (бит) команды LEND(810) или LEND (810) NOT не переводится в состояние ON или OFF (соответственно), или до тех пор, пока условие выполнения команды LEND(810) не перейдет в состояние ON.

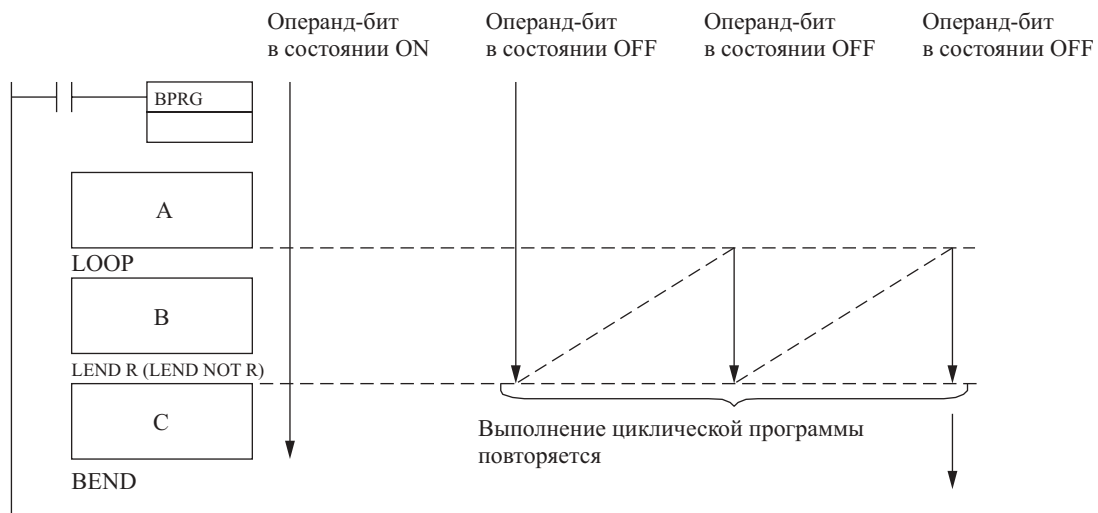
#### Применение условия выполнения для команды LEND(810)

Команда LEND(809) может программироваться с использованием или без использования операнда (бита). При отсутствии операнда перед командой LEND(809) должно создаваться условие выполнения, начиная командой LD. Если условие выполнения находится в состоянии OFF, выполнение циклической программы начинается с команды, следующей за командой LOOP(809). Если условие выполнения находится в состоянии ON, выполнение циклической программы завершается, и программа переходит к выполнению команды, следующей за командой LEND(810).



#### Применение операнда-бита для команд LEND(810) или LEND (810) NOT

Обе команды LEND(810) и LEND (810) NOT могут программироваться с использованием или без использования операнда (бита). Если операнд команды LEND(810) находится в состоянии OFF (или в состоянии ON для команды LEND(810) NOT), выполнение циклической программы начинается с команды, следующей за командой LOOP(809). Если операнд команды LEND(810) находится в состоянии ON (или в состоянии OFF для команды LEND(810) NOT), выполнение циклической программы завершается, и программа переходит к выполнению команды, следующей за командой LEND(810).



**Примечание:** Состояние операнда для команды *LEND(810) NOT* инвертируется.

**Примечание:** 1. При выполнении циклической программы регенерация ввода/вывода не производится. При необходимости выполнения регенерации ввода/вывода используйте команду *IORF(184)*.

2. При достаточно длительном выполнении циклической программы, длительность цикла может превысить заданное значение. Создавайте программу таким образом, чтобы заданное максимальное время цикла не превышалось.

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, если команда управления циклом находится вне блочной программы. OFF в других случаях.
Флаг равенства	=	OFF
Флаг отрицательного значения.	N	OFF

#### Меры предосторожности

Циклические программы не могут размещаться внутри других циклических программ.

Неправильно  
 LOOP(809)  
 LOOP(809)  
 LEND(810)  
 LEND(801)

Не нарушайте порядок следования команд LOOP и LEND.

Неправильно  
 LEND(810)  
 ...  
 LOOP(809)

Условные разветвления могут использоваться внутри циклической программы. В этом случае все операции, выполняемые в ответвлении должны находиться в данной циклической программе.

Правильно	Неправильно
LOOP(809)	LOOP(809)
IF9802)	IF(802)
IF9802)	IF(802)
IEND(804)	IEND(804)
IEND(804)	LEND(810)
LEND(810)	IEND(804)

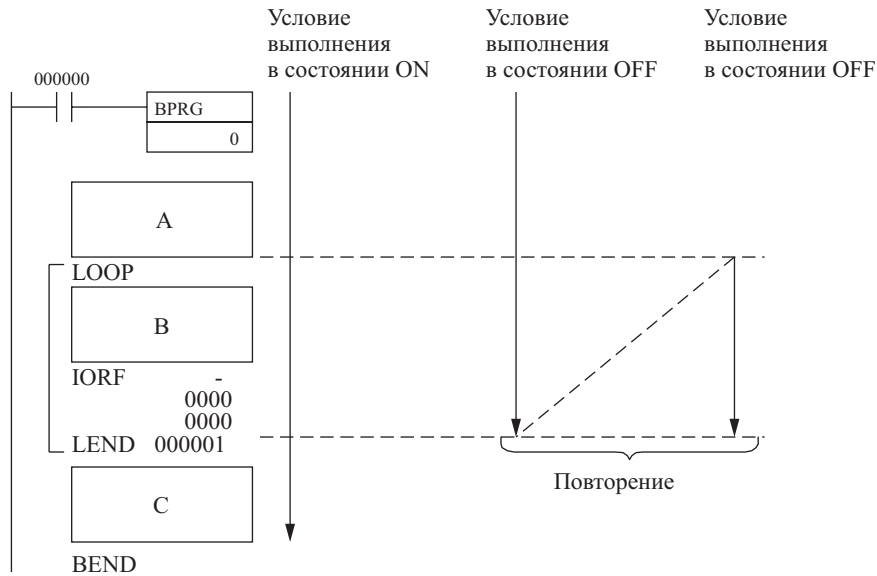
Если команда LOOP(809) не выполняется, выполняется команда NOP.

Если команда управления циклом находится вне блочной программы, определяется ошибка и флаг ошибки переводится в состояние ON.

При выполнении команды управления циклом флаг равенства и флаг отрицательного значения переводятся в состояние OFF.

### Примеры

В следующем ниже примере, когда CIO 000000 находится в состоянии ON, производится выполнение блочной программы. После завершения выполнения «А», «В» и IORF(184) выполняются неоднократно, до тех пор, пока CIO 000001 не будет переведен в состояние ON. В этот момент выполняется С и выполнение блочной программы завершается.



Адрес	Команда	Операнд
000220	LD	000000
intbl 000221	BPRG	0
...	A	...
000210	LOOP	—
	B	
000220	IORF	
		0000
...		0000
000221	LEND	000001
	C	
000220	BEND	—

## 3-31 Команды обработки текстовой строки

В настоящем разделе приводится описание команд, используемых для управления текстовой строкой.

Команда	Мнемоническое обозначение	Функциональный код	Страница
MOV STRING	MOV\$	664	710
CONCATENATE STRING	+\$	656	712
GET STRING LEFT	LEFT\$	652	714
GET STRING RIGHT	RGHT\$	653	716
SET STRIMG MIDDLE	MID\$	654	719
FIND IN STRING	FIND\$	660	721
STRING LENGTH	LEN\$	650	723
REPLACE IN STRING	RPLC\$	661	725
DELETE STRING	DEL\$	658	727
EXCHANGE STRING	XCHG\$	665	730
CLEAR STRING	CLR\$	666	732
INSERT INTO STRING	INS\$	657	733
String Comparison Instructions	=\$, <>\$, <\$, <=\$, >\$, >=\$	670...675	736

## 3-31-1 Обзор операций управления текстовой строкой

Данные, начиная с начала до кода нуля (00 шестн.) обрабатываются в виде текстовой строки, выраженной в коде ASCII (кроме одного байта, занимаемого специальными символами). Данные запоминаются от старших байтов до младших байтов и от старших слов к младшим словам.

Когда в строке содержится нечетное количество символов, в младшем байте последнего слова записывается код нуля (00).

Пример: текстовая строка ABCDE

A	→	B		41	42
C	←	D		43	44
E	←	NUL		45	00

Когда в строке содержится четное количество символов, в младшем байте и старшем байте слова, следующего за последним словом, записывается код нуля (00).

Пример: текстовая строка ABCDE

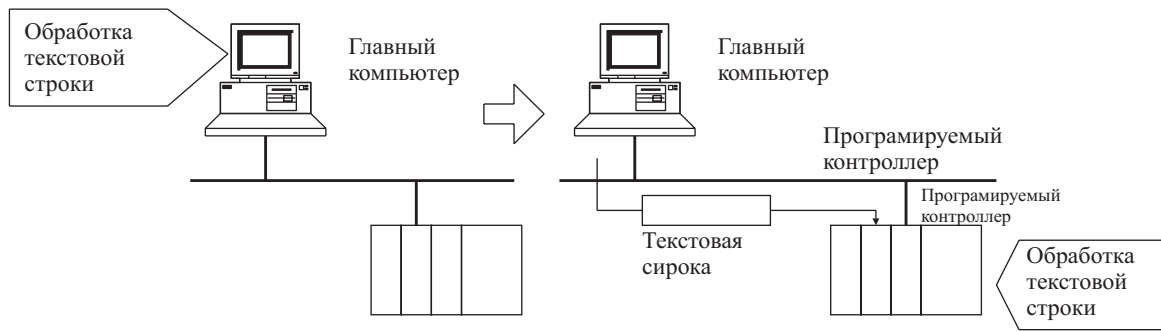
A	→	B		41	42
C	←	D		43	44
NUL	←	NUL		00	00

Как показано на следующем ниже рисунке, текстовая строка может задаваться путем указания только первого слова строки. В этом случае данные в виде текстовой строки вплоть до следующего кода нуля (00), будут обрабатываться в виде одного блока данных в ASCII коде.

Пример: MOV\$ D00000 D00100

D00000	41	42		D00100	41	42
D00001	43	44	→	D00101	43	44
D00002	45	NUL		D00102	45	00

Команды обработки текстовой строки могут применяться для выполнения в Программируемом контроллере различных операций с текстовой строкой (например, данные об изделии и т.д.), которая используется в главном компьютере.



**Символы ASCII кода**

В следующей ниже таблице приведены коды ASCII, которые могут обрабатываться командами, описываемыми в данном разделе.

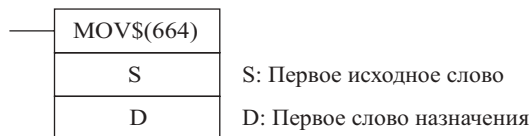
Четыре правых бита	Четыре левых бита															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			Sp	0	@	P	`	p								
1			!	1	A	Q	a	q								
2			"	2	B	R	b	r								
3			#	3	C	S	c	s								
4			\$	4	D	T	d	t								
5			%	5	E	U	e	u								
6			&	6	F	V	f	v								
7			'	7	G	W	g	w								
8			(	8	H	X	h	x								
9			)	9	I	Y	i	y								
A			*	:	J	Z	j	z								
B			+	;	K	[	k	{								
C			,	<	L		l									
D			-	=	M	]	m	}								
E			.	>	N	^	n	~								
F			/	?	O	_	o									

**3-31-2 Команда перемещения строки MOV STRING:MOV\$(664)**

**Назначение**

С помощью данной команды осуществляется передача текстовой строки.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	MOV&(664)
	Выполняется один раз для дифференцирования вверх	@ MOV&(664)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****S: Первое исходное слово**

	15	0
S	максимум 4095 знаков +NUL	
до		
S+максимум 2047 слов		

**D: Первое слово назначения**

	15	0
D	максимум 4095 знаков +NUL	
до		
D+максимум 2047 слов		

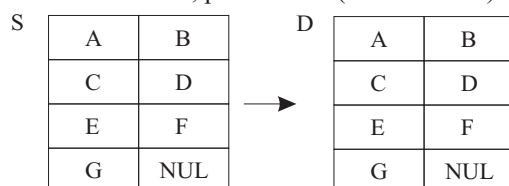
- Примечание:**
1. Данные, содержащиеся в словах S S+максимальное количество слов (2047) и D D+максимальное количество слов (2047), должны находиться в одной области данных.
  2. Данные, содержащиеся в словах S...S+максимальное количество слов (2047) и D...D+максимальное количество слов (2047), могут перекрываться.

**Спецификации операндов**

Область	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A000...A447 A448...A959	A448...A959
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n=0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)	
Константы	-	
Регистры данных	-	
Индексные регистры	-	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)... ,IR15+(++) ,(- -)IR0...,(- -)IR15	

**Описание**

По команде MOV\$(664) производится передача данных текстовой строки, указанной в S, в том виде как она есть (включая завершающий код (00)), в D. Наибольшее количество символов, которое может указываться словом S, равно 4095 (0FFF шестн.).



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда словом S указывается более 4095 символов. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда в D передается 0000 (шестн.). OFF в других случаях.

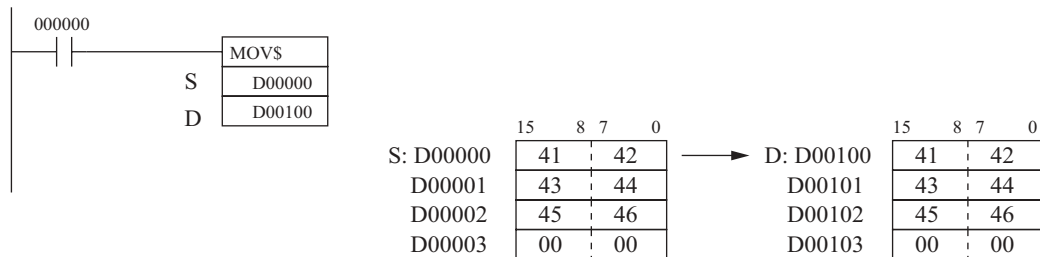
**Меры предосторожности**

В случае, когда словом S указывается более 4095 символов, определяется ошибка и флаг ошибки переводится в состояние ON.

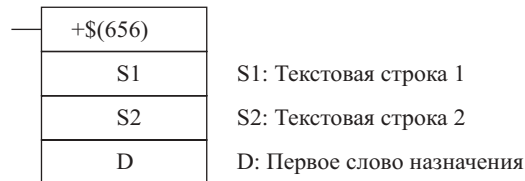
Если в слово D передается 0000 (шестн.) флаг равенства переводится в состояние ON.

**Пример**

В данном примере команда MOV\$(664) используется для передачи текстовой строки ABCDEF.

**3-31-3 Команда соединения строк CONCATENATE STRING: +\$(656)****Назначение**

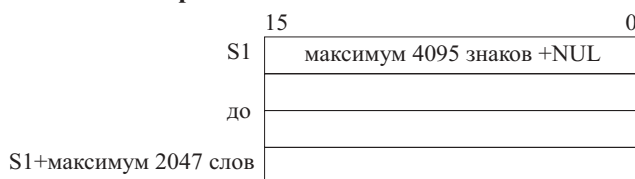
С помощью данной команды осуществляется подсоединение одной текстовой строки к другой текстовой строке.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	+\$(656)
	Выполняется один раз для дифференцирования вверх	@ +\$(656)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****S1: Текстовая строка 1**

**S2: Текстовая строка 2**

	15	0
S2	максимум 4095 знаков +NUL	
до		
S2+максимум 2047 слов		

**D: Первое слово назначения**

	15	0
D	максимум 4095 знаков +NUL	
до		
D+максимум 2047 слов		

- Примечание:**
1. Данные, содержащиеся в словах S1...S1+максимальное количество слов (2047), S2...S2+максимальное количество слов (2047) и D...D +максимальное количество слов (2047), должны находиться в одной области данных.
  2. Данные, содержащиеся в словах S2...S2+максимальное количество слов (2047) и D...D +максимальное количество слов (2047), перекрываться не должны.

**Спецификации операндов**

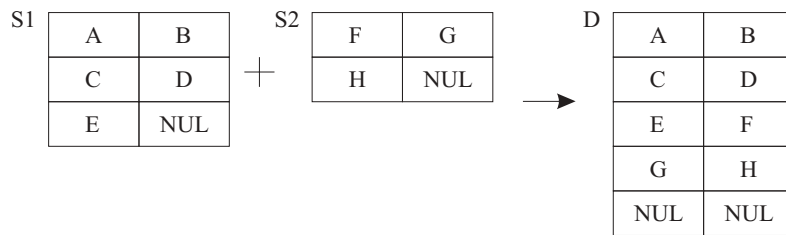
Область	S1	S2	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A447 A448...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n=0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)		
Константы	-		
Регистры данных	-		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048 +2047,IR0...-2048...+2047,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15		

**Описание**

По команде +\$(664) производится соединение данных текстовой строки, указанной в S1, к данным текстовой строки, указанной в S2. Результат выводится в D в виде текстовой строки (включая код окончания 00). Наибольшее количество символов, которое может указываться словами S1 и S2, является 4095 (0FFF шестн.). Если после указанных 4095 символов отсутствует код окончания 00, определяется ошибка и флаг ошибки переводится в состояние ON. Более того, результирующая строка не должна содержать более 4095 символов (0FFF шестн.). Если результирующая строка содержит большее количество символов, в D записываются только первые 4095 символов плюс код нуля, записываемый дополнительно (т.е. всего 4096 символов).

В случае, когда в строках S1 и S2 содержатся нули, в слово назначения D записывается два нуля (0000 шестн.).





**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда словами S1 и S2 указывается более 4095 символов. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда в D передается 0000 (шестн.). OFF в других случаях.

**Меры предосторожности**

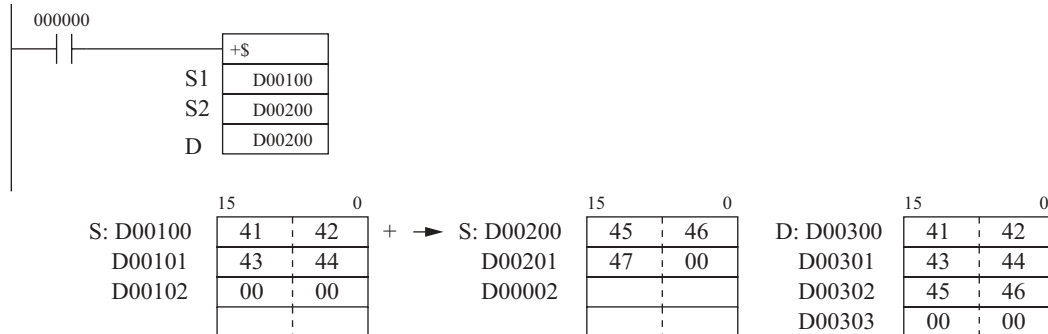
В случае, когда словами S1 и S2 указывается более 4095 символов, определяется ошибка и флаг ошибки переводится в состояние ON.

Если в слово D передается 0000 (шестн.) флаг равенства переводится в состояние ON.

Не допускайте перекрытия первого слова, указанного в D, областью данных, предназначенной для S2. В случае перекрытия областей данных команда не будет выполняться должным образом.

**Пример**

В данном примере команда +&(656) используется для соединения текстовой строки ABCD и текстовой строки EF. Результат записывается в D.

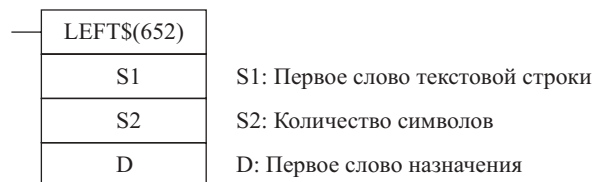


**3-31-4 Команда выборки данных из строки GET STRING LEFT: LEFT\$(652)**

**Назначение**

С помощью данной команды осуществляется выборка указанного количества символов, находящихся в начале текстовой строки.

**Символ релейно-контактной схемы**



**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	LEFT\$(652)
	Выполняется один раз для дифференцирования вверх	@ LEFT\$(652)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****S1: Текстовая строка**

	15	0
S1	максимум 4095 знаков +NUL	
до		
S1+максимум 2047 слов		

**S2: Количество символов (0000 0FFF или &0 4095)**

	15	0
S2	максимум 4095 знаков +NUL	
до		
S2+максимум 2047 слов		

**D: Первое слово назначения**

	15	0
D	максимум 4095 знаков +NUL	
до		
D+максимум 2047 слов		

- Примечание:**
1. Данные, содержащиеся в словах S1...S1+максимальное количество слов (2047) и D...D+максимальное количество слов (2047), должны находиться в одной области данных.
  2. Данные, содержащиеся в словах S1...S1+максимальное количество слов (2047) и D...D+максимальное количество слов (2047), могут перекрываться.

**Спецификации операндов**

Область	S1	S2	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A447 A448...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n=0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)		
Константы	—	#0000...#FFFF (двоичные) или &0...&4095	—
Регистры данных	—		
Индексные регистры	—		

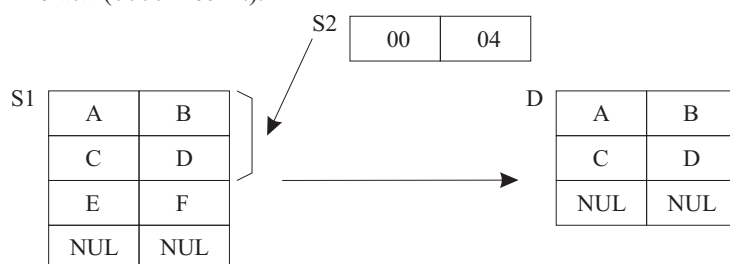
Область	S1	S2	D
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048 +2047,IR0...-2048...+2047,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++)  ,-(-)IR0...,-(-)IR15		

**Описание**

По команде LEFT\$(652) производится чтение заданного в S2 количества символов с начала первого слова текстовой строки, указанного словом S1. Результат заносится в D (в конце данных прибавляется 00).

Если количество выбранных слов превышает указанное в S1 количество символов, вся текстовая строка S1 выводится в D.

Если в качестве количества символов, подлежащих чтению, задается 0 (0000 шестн.), в D заносится значение «0» (0000 шестн.).

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда словом S1 указывается более 4095 символов. Переводится в состояние ON, когда словом S2 указывается более 4095 символов (0FFF). OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда в D передается 0000 (шестн.). OFF в других случаях.

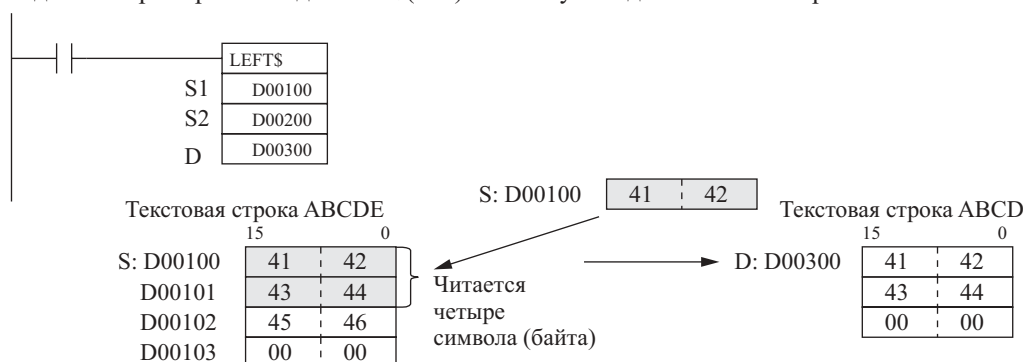
**Меры предосторожности**

Максимальное количество символов, подлежащих чтению и указываемое в S2, не должно превышать 4095 (шестн.). В случае, когда словом S указывается более 4095 символов, определяется ошибка и флаг ошибки переводится в состояние ON.

Если в слово D передается 0000 (шестн.) флаг равенства переводится в состояние ON.

**Пример**

В данном примере команда LEFT\$(652) используется для чтения четырех символов.

**3-31-5 Команда выборки данных из строки GET STRING RIGHT: RIGHT\$(653)****Назначение**

С помощью данной команды осуществляется выборка указанного количества символов, находящихся в конце текстовой строки.

**Символ релейно-контактной схемы**

RGHT\$(653)	
S1	S1: Первое слово текстовой строки
S2	S2: Количество символов
D	D: Первое слово назначения

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	RIGHT\$(653)
	Выполняется один раз для дифференцирования вверх	@ RIGHT\$(653)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****S1: Текстовая строка**

	15	0
S1	максимум 4095 знаков +NUL	
до		
S1+максимум 2047 слов		

**S2: Количество символов (0000 0FFF или &0 4095)**

	15	0
S2	максимум 4095 знаков +NUL	
до		
S2+максимум 2047 слов		

- Примечание:**
- Данные, содержащиеся в словах S1...S1+максимальное количество слов (2047) и D...D+максимальное количество слов (2047), должны находиться в одной области данных.
  - Данные, содержащиеся в словах S1...S1+максимальное количество слов (2047) и D...D+максимальное количество слов (2047), могут перекрываться.

**Спецификации операндов**

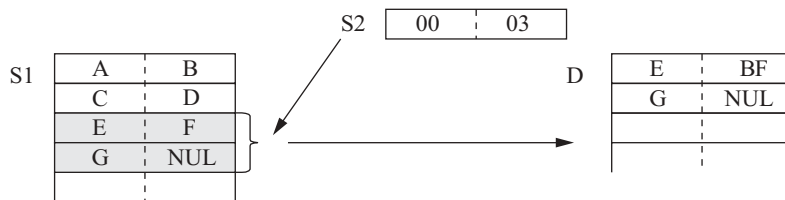
Область	S1	S2	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A447 A448...A959	A448...A959	
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n=0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)		

Область	S1	S2	D
Константы	—	#0000...#FFFF (двоичные) или &0...&4095	—
Регистры данных	—		
Индексные регистры	—		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048+2047,IR0...-2048...+2047,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...IR15(++) ,-( -)IR0...,-( -)IR15		

**Описание**

По команде RIGHT\$(653) производится чтение заданного в S2 количества символов, находящихся с правой стороны строки, указанной в S1. Чтение начинается со слова текстовой строки, определяемого количеством символов, подлежащих чтению, и заканчивается кодом окончания (00). Результат заносится в D (в конце данных прибавляется 00).

Если в качестве количества символов, подлежащих чтению, задается 0 (0000 шестн.), в D заносится значение «0» (0000 шестн.).



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда словом S1 указывается более 4095 символов. Переводится в состояние ON, когда словом S2 указывается более 4095 символов (0FFF). OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда в D передается 0000 (шестн.). OFF в других случаях.

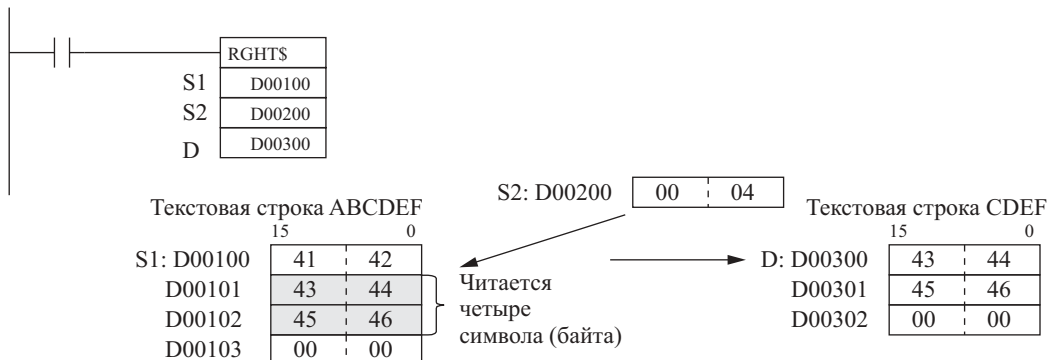
**Меры предосторожности**

Максимальное количество символов, подлежащих чтению и указываемое в S2, не должно превышать 4095 (шестн.). В случае, когда словом S2 указывается более 4095 символов, определяется ошибка и флаг ошибки переводится в состояние ON.

Если в слово D передается 0000 (шестн.) флаг равенства переводится в состояние ON.

**Пример**

В данном примере команда RIGHT\$(653) используется для чтения четырех символов.



## 3-31-6 Команда выборки данных из середины строки GET STRING MIDDLE: MID\$(654)

**Назначение**

С помощью данной команды осуществляется выборка указанного количества символов, находящихся в любом месте в середине текстовой строки.

**Символ релейно-контактной схемы**

MID\$(654)	
S1	S1: Первое слово текстовой строки
S2	S2: Количество символов
S3	S3: Начальная позиция
D	D: Первое слово назначения

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	MID\$(654)
	Выполняется один раз для дифференцирования вверх	@ MID\$(654)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****S1: Текстовая строка**

	15	0
S1	максимум 4095 знаков +NUL	
до		
S1+максимум 2047 слов		

**S2: Количество символов (0000 0FFF или &0 4095)****S3: Начальная позиция (0001 0FFF или &0 4095)****D: Первое слово назначения**

	15	0
D	максимум 4095 знаков +NUL	
до		
D+максимум 2047 слов		

- Примечание:**
1. Данные, содержащиеся в словах S1...S1+максимальное количество слов (2047) и D...D+максимальное количество слов (2047), должны находиться в одной области данных.
  2. Данные, содержащиеся в словах S1...S1+максимальное количество слов (2047) и D...D+максимальное количество слов (2047), могут перекрываться.

**Спецификации операндов**

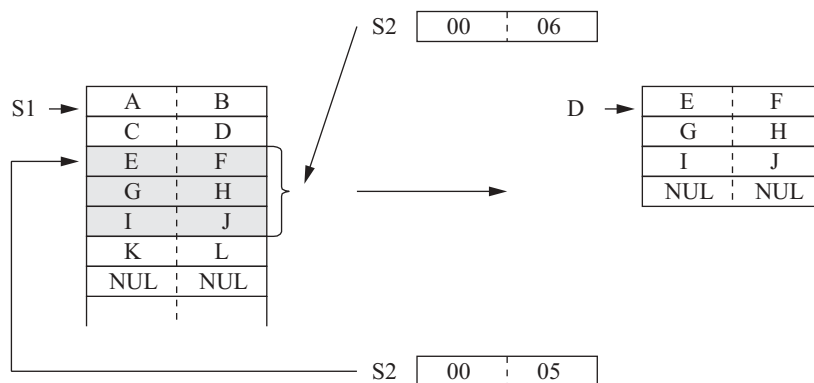
Область	S1	S2	S3	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143			
Рабочая область	W000...W511			
Область удержания бита (Holding bit area)	H000...H511			
Область вспомогательных битов	A000...A447 A448...A959			A448...A959
Область таймера	T0000...T4095			
Область счетчика	C0000...C4095			

Область	S1	S2	S3	D
Область DM	D00000...D32767			
Область EM, не содержащая банков	E00000...E32767			
Область EM, содержащая банки	En_00000...En_327/67 (n=0...C)			
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000 @En_32767 (n=0...C)			
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)			
Константы	–	#0000...#FFFF (двоичные) или &0...&4095	#0001...#FFFF (двоичные) или &0...&4095	–
Регистры данных	–	DR0...DR15		
Индексные регистры	–			
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-(-)IR15			

### Описание

По команде MID\$(654) производится чтение заданного в S2 количества символов с начала слова, указанного в S3. Текстовая строка указывается первым словом, указанным в S1. Чтение данных заканчивается кодом окончания (00). Результат заносится в D (в конце данных прибавляется 00).

Если количество слов, подлежащих чтению, превышает длину текстовой строки, указанной в S1, вся текстовая строка S1 выводится в D.



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда словом S1 указывается более 4095 символов. Переводится в состояние ON, когда словом S2 указывается более 4095 символов (0FFF). Переводится в состояние ON, когда данные S3 находятся в пределах от 1 до 4095 (от 0001 до 0FFF). Переводится в состояние ON, когда S3 больше S1. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда в D передается 0000 (шестн.). OFF в других случаях.

### Меры предосторожности

Диапазон задания начальной позиции в S2 – от 1-го до 4095-го символа (от 0001 до 0FFF). Если задаваемое значение выходит за указанные пределы определяется ошибка и флаг ошибки переводится в состояние ON.

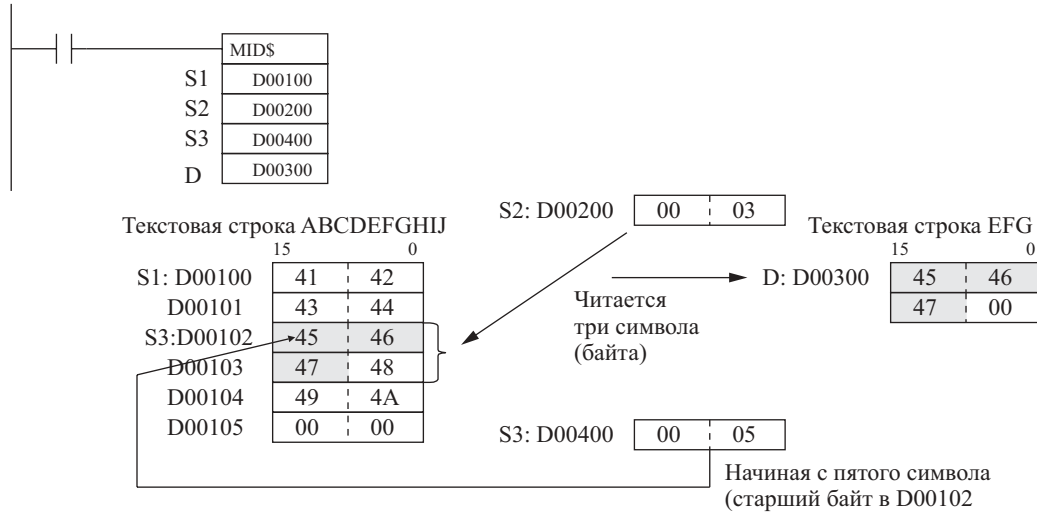
Максимальное количество символов, подлежащих чтению и указываемое в S2, не должно превышать 4095 (шестн.). В случае, когда словом S2 указывается более 4095 символов, определяется ошибка и флаг ошибки переводится в состояние ON.

Если в качестве количества символов, подлежащих чтению, указывается 0 (0000 шестн.), в D заносится два символа нуля (0000 шестн.).

Если в слово D передается 0000 (шестн.) флаг равенства переводится в состояние ON.

**Пример**

В данном примере команда MID\$(654) используется для чтения четырех символов.



3-31-7 Команда нахождения заданной текстовой строки в текстовой строке FIND IN STRING: FIND\$(660)

**Назначение**

С помощью данной команды осуществляется нахождение заданной текстовой строки в текстовой строке.

**Символ релейно-контактной схемы**

FIND\$(660)	
S1	S1: Первое слово исходной текстовой строки
S2	S2: Первое слово искомой текстовой строки
D	D: Первое слово назначения

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	FIND\$(660)
	Выполняется один раз для дифференцирования вверх	@ FIND\$(660)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды**

**S1: Исходная текстовая строка**

S1	15	0
до	максимум 4095 знаков +NUL	
S1+максимум 2047 слов		



**S2: Искомая текстовая строка**

	15	0
S2	максимум 4095 знаков +NUL	
до		
S2+максимум 2047 слов		

**Примечание:** Данные, содержащиеся в словах S1...S1+максимальное количество слов (2047) и S2...S2+максимальное количество слов (2047), должны находиться в одной области данных.

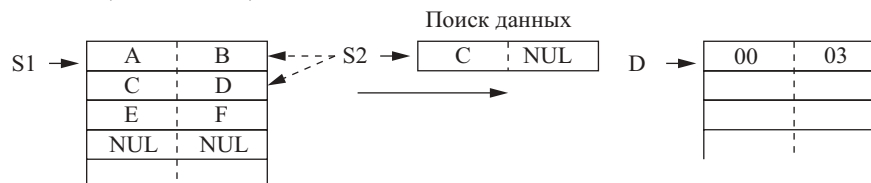
**Спецификации операндов**

Область	S1	S2	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143		
Рабочая область	W000...W511		
Область удержания бита (Holding bit area)	H000...H511		
Область вспомогательных битов	A000...A447 A448...A959		A448...A959
Область таймера	T0000...T4095		
Область счетчика	C0000...C4095		
Область DM	D00000...D32767		
Область EM, не содержащая банков	E00000...E32767		
Область EM, содержащая банки	En_00000...En_32767 (n=0...C)		
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)		
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)		
Константы	-		
Регистры данных	-		
Индексные регистры	-		
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048+2047,IR0...-2048...+2047,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15		

**Описание**

По команде FIND\$(660) производится поиск текстовой строки, указанной в S2, и вывод результата (число символов от начала S1). Результат заносится в D в виде двоичных данных. Если в данной строке отсутствует искомая строка, в D заносится значение 0000 шестн.

Если в качестве количества символов, подлежащих чтению, задается 0 (0000 шестн.), в D заносится значение «0» (0000 шестн.).

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда словом S1 или S2 указывается более 4095 символов. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда в D передается 0000 (шестн.). OFF в других случаях.

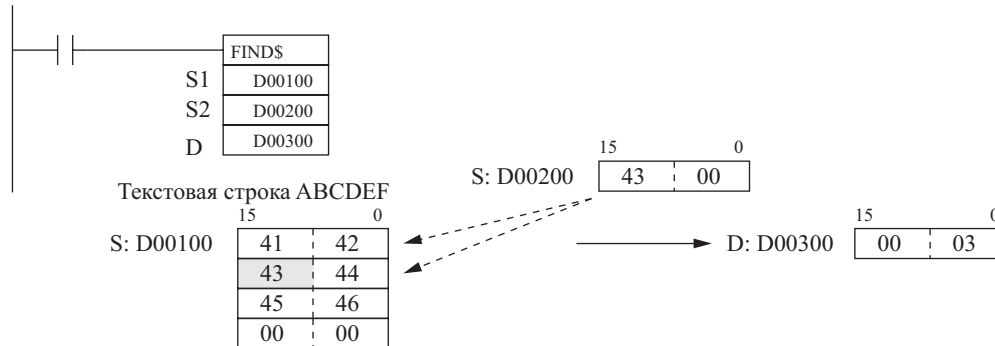
**Меры предосторожности**

Максимальное количество символов, подлежащих чтению и указываемое в S1 или S2, не должно превышать 4095 (шестн.). В случае, когда указывается более 4095 символов, определяется ошибка и флаг ошибки переводится в состояние ON.

Если в слово D передается 0000 (шестн.) флаг равенства переводится в состояние ON.

**Пример**

В данном примере команда FIND\$(660) используется для поиска в текстовой строке одного символа.

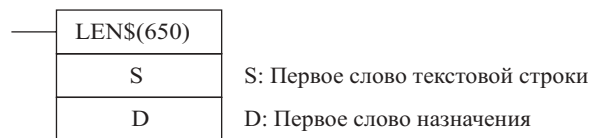


**3-31-8 Команда вычисления длины текстовой строки STRING LENGTH: LEN\$(650)**

**Назначение**

С помощью данной команды осуществляется вычисление длины текстовой строки.

**Символ релейно-контактной схемы**



**Модификации**

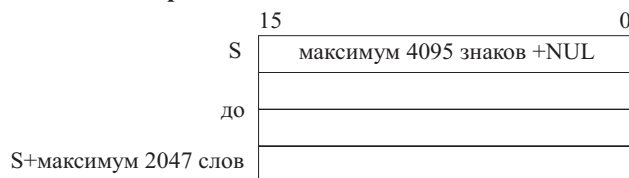
Модификации	Выполняется в каждом цикле для состояния ON	LEN\$(650)
	Выполняется один раз для дифференцирования вверх	@ LEN\$(650)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды**

**S:** Текстовая строка



**Примечание:** Данные, содержащиеся в словах S1...S1+максимальное количество слов (2047) и D...D+максимальное количество слов (2047), должны находиться в одной области данных.

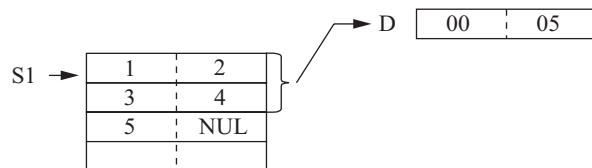
**Спецификации операндов**

Область	S	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	

Область	S	D
Область вспомогательных битов	A000...A447 A448...A959	A448...A959
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n=0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)	
Константы	—	DR0...DR15
Регистры данных	—	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048 +2047,IR0...-2048...+2047,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,-(- -)IR15	

**Описание**

По команде RIGHTS\$(653) производится вычисление количества символов, начиная с первого слова текстовой строки, указанного словом S1, и заканчивая кодом окончания (00) включительно. Результат заносится в D в виде двоичных данных. Если в начале текстовой строки находится 0, результатом вычисления будет 0000 шестн.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, если результат вычисления превышает значение, равное 4095 символов. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда результат вычисления равен нулю. OFF в других случаях.

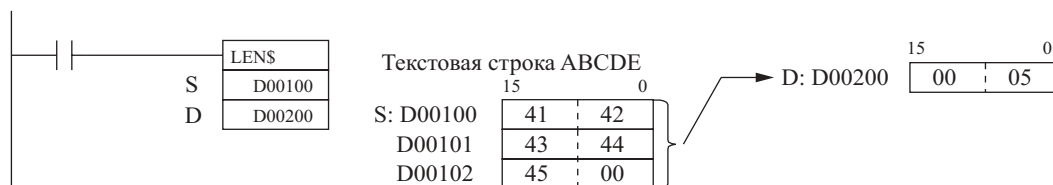
**Меры предосторожности**

Максимальное количество символов не должно превышать 4095 (0FFF шестн.). В случае, когда словом S2 указывается более 4095 символов (т.е. после 4095 символов отсутствует код нуля 00 шестн.), определяется ошибка и флаг ошибки переводится в состояние ON.

Если в слово D передается 0000 (шестн.) флаг равенства переводится в состояние ON.

**Пример**

В данном примере команда LENS\$(650) используется для вычисления количества символов и вывода результата в D.



### 3-31-9 Команда замещения текстовой строки указанной текстовой строкой REPLACE IN STRING: RPLC\$(661)

#### Назначение

С помощью данной команды осуществляется замещение текстовой строки указанной текстовой строкой, начиная с указанной позиции.

#### Символ релейно-контактной схемы

RPLC\$(661)	
S1	S1: Первое слово текстовой строки
S2	S2: Первое слово замещающей текстовой строки
S3	S3: Количество символов
S4	S4: Начальная позиция
D	D: Первое слово назначения

#### Модификации

Модификации	Выполняется в каждом цикле для состояния ON	RPLC\$(661)
	Выполняется один раз для дифференцирования вверх	@ RPLC\$(661)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

##### S1: Текстовая строка

	15	0
S1	максимум 4095 знаков +NUL	
до		
S1+максимум 2047 слов		

##### S2: Замещающая текстовая строка

	15	0
S2	максимум 4095 знаков +NUL	
до		
S2+максимум 2047 слов		

##### S3: Количество символов (0000 0FFF или &0 4095)

##### S4: Начальная позиция (0001 0FFF или &0 4095)

##### D: Первое слово назначения

	15	0
D	максимум 4095 знаков +NUL	
до		
D+максимум 2047 слов		

**Примечание:** 1. Данные, содержащиеся в словах S1...S1+максимальное количество слов (2047), S2...S2+максимальное количество слов (2047) и D...D +максимальное количество слов (2047), должны находиться в одной области данных.

2. Данные от D до D +максимальное количество слов (2047) и, или от S1 до S1+максимальное количество слов (2047), или от S2 до S2 + максимальное количество слов (2047), могут перекрываться.

### Спецификации операндов

Область	S1	S2	S3	S4	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143				
Рабочая область	W000...W511				
Область удержания бита (Holding bit area)	H000...H511				
Область вспомогательных битов	A000...A447 A448...A959				448...A959
Область таймера	T0000...T4095				
Область счетчика	C0000...C4095				
Область DM	D00000...D32767				
Область EM, не содержащая банков	E00000...E32767				
Область EM, содержащая банки	En_00000...En_327/67 (n=0...C)				
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)				
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)				
Константы	-		#0000...#FFF F (двоичные) или &0...&4095	#0001...#FFF F (двоичные) или &0...&4095	-
Регистры данных	-				
Индексные регистры	-				
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048 +2047,IR0...-2048...+2047,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...,IR15+(++) ,(- -)IR0...,(- -)IR15				

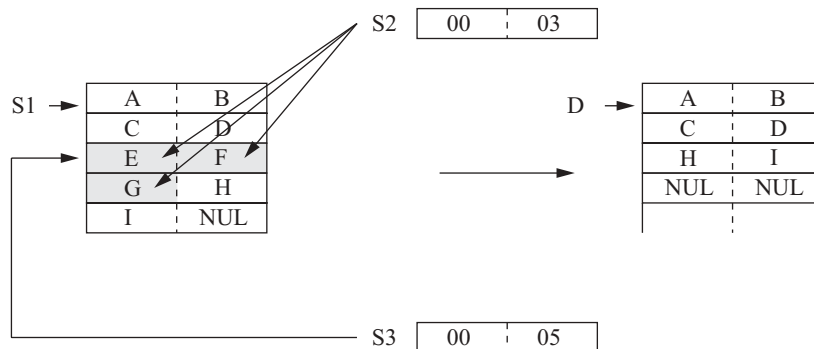
### Описание

По команде RPLC\$(661) производится замена части текстовой строки, указанной в S1, текстовой строкой, указанной в S2. Начальная позиция для осуществления замены указывается в S4. Результат заносится в D в виде текстовой строки (в конце данных прибавляется 00). Количество замещаемых символов указывается в S3.

Максимальное количество символов результата – 4095 (0FFF шестн.). Если количество символов превышает указанное значение, в D заносится только 4095 символов плюс символ нуля.

При помощи данного действия может замещаться от 0 до 4095 символов (0000...0FFF шестн.). Если количество равно нулю, текстовая строка, указанная в S1 заносится в D без изменения. Если текстовая строка S2 равна нулю, выполняемое действие будет аналогично удалению указанного диапазона данных из строки S1.

В случае, когда текстовая строка S1 заполняется от начала до конца символами нуля, в D заносится два символа нуля (0000 шестн.).



**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда словом S1 указывается более 4095 символов. Переводится в состояние ON, когда словом S3 указывается более 4095 символов (0FFF). Переводится в состояние ON, когда данные S4 находятся в пределах от 1 до 4095 (от 0001 до 0FFF). OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда в D передается 0000 (шестн.). OFF в других случаях.

**Меры предосторожности**

Максимальное количество символов для S1 и S2 – 4095 (0FFF шестн.). Если задаваемые значения выходят за указанный предел (т.е. после 4095 символов отсутствует символ нуля), определяется ошибка и флаг ошибки переводится в состояние ON.

Диапазон задания начальной позиции в S4 – от 1-го до 4095-го символа (от 0001 до 0FFF). Если установка выходит за указанные пределы, определяется ошибка и флаг ошибки переводится в состояние ON.

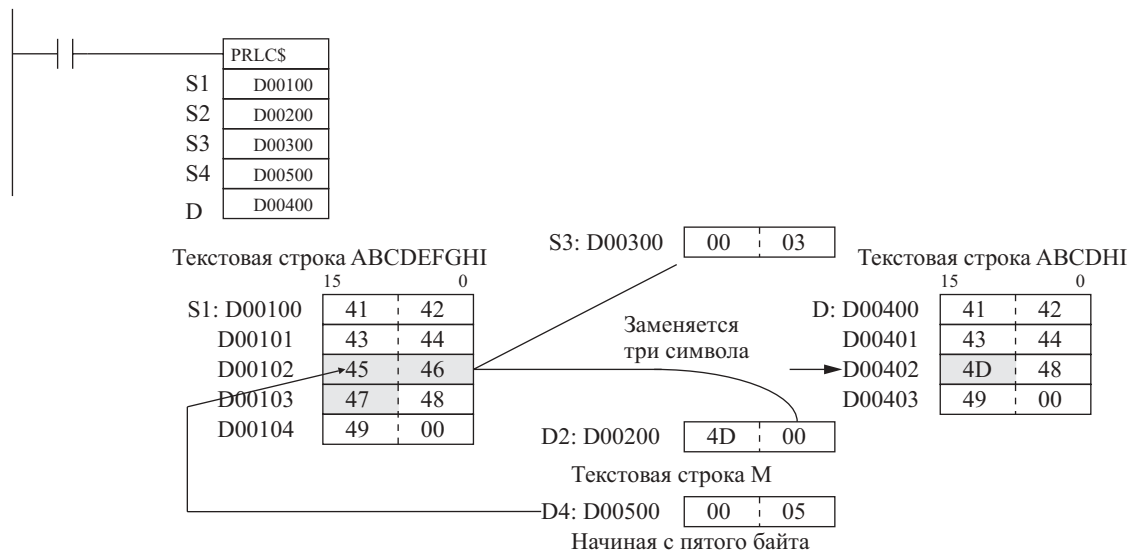
Если начальная позиция, указанная в S4, превышает величину текстовой строки, указанной в S1, определяется ошибка и флаг ошибки переводится в состояние ON.

Если в слово D передается 0000 (шестн.) флаг равенства переводится в состояние ON.

Устанавливайте значение первого слова назначения D таким образом, чтобы оно не перекрывало области замещающей строки S2. Если происходит перекрытие, команда RPLC\$(654) не может функционировать корректно.

**Пример**

В данном примере команда RPLC\$(654) используется для чтения трех символов.

**3-31-10 Команда удаления текстовой строки DELETE STRING: DEL\$(658)****Назначение**

С помощью данной команды осуществляется удаление указанной текстовой строки из середины текстовой строки.

**Символ релейно-контактной схемы**

DEL\$(658)	
S1	S1: Первое слово текстовой строки
S2	S2: Количество символов
S3	S3: Начальная позиция
D	D: Первое слово назначения

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	DEL\$(658)
	Выполняется один раз для дифференцирования вверх	@ DEL\$(658)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****S1: Текстовая строка**

	15	0
S1	максимум 4095 знаков +NUL	
до		
S1+максимум 2047 слов		

**S2: Количество символов (0000 0FFF или &0 4095)****S3: Начальная позиция (0001 0FFF или &0 4095)****D: Первое слово назначения**

	15	0
D	максимум 4095 знаков +NUL	
до		
D+максимум 2047 слов		

- Примечание:**
- Данные, содержащиеся в словах S1...S1+максимальное количество слов (2047), S2...S2+максимальное количество слов (2047) и D...D+максимальное количество слов (2047), должны находиться в одной области данных.
  - Данные от S1 до S1+максимальное количество слов (2047) и от D до D+максимальное количество слов (2047) могут перекрываться.

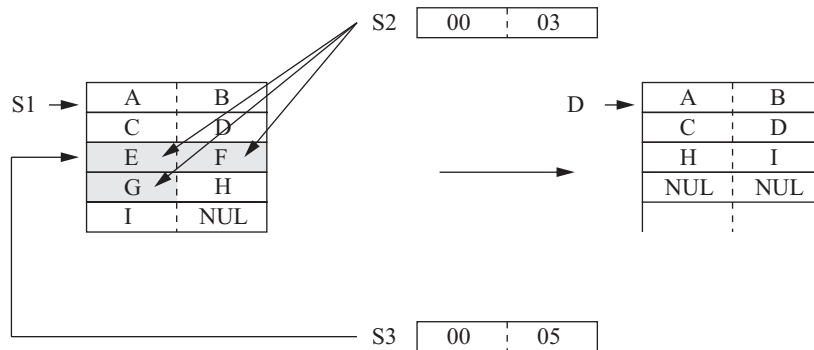
**Спецификации операндов**

Область	S1	S2	S3	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143			
Рабочая область	W000...W511			
Область удержания бита (Holding bit area)	H000...H511			
Область вспомогательных битов	A000...A447 A448...A959			A448...A959
Область таймера	T0000...T4095			
Область счетчика	C0000...C4095			
Область DM	D00000...D32767			
Область EM, не содержащая банков	E00000...E32767			
Область EM, содержащая банки	En_00000...En_327/67 (n=0...C)			
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)			
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)			
Константы	–	#0000...#FFFF (двоичные) или &0...&4095	#0001...#FFF F (двоичные) или &0...&4095	–
Регистры данных	–	DR0...DR15	–	
Индексные регистры	–			

Область	S1	S2	S3	D
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048...+2047 ,IR0 -2048...+2047 ,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,-(-)IR0...,-(-)IR15			

**Описание**

По команде DEL\$(658) производится удаление заданного в S2 количества символов из текстовой строки, указанной в S1. Начальное слово указывается в S3. Результат заносится в D в виде текстовой строки (в конце данных прибавляется 00).

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда словом S1 указывается более 4095 символов. Переводится в состояние ON, когда словом S2 указывается более 4095 символов (0FFF шестн.). Переводится в состояние ON, когда данные S3 находятся в пределах от 1 до 4095 (от 0001 до 0FFF шестн.). Переводится в состояние ON, когда S3 больше S1. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда в D передается 0000 (шестн.). OFF в других случаях.

**Меры предосторожности**

Максимальное количество символов для S1– 4095 (0FFF шестн.). Если задаваемое значение выходит за указанный предел (т.е. после 4095 символов отсутствует символ нуля), определяется ошибка и флаг ошибки переводится в состояние ON.

Диапазон задания начальной позиции в S3 – от 1-го до 4095-го символа (от 0001 до 0FFF). Если установка выходит за указанные пределы, определяется ошибка и флаг ошибки переводится в состояние ON.

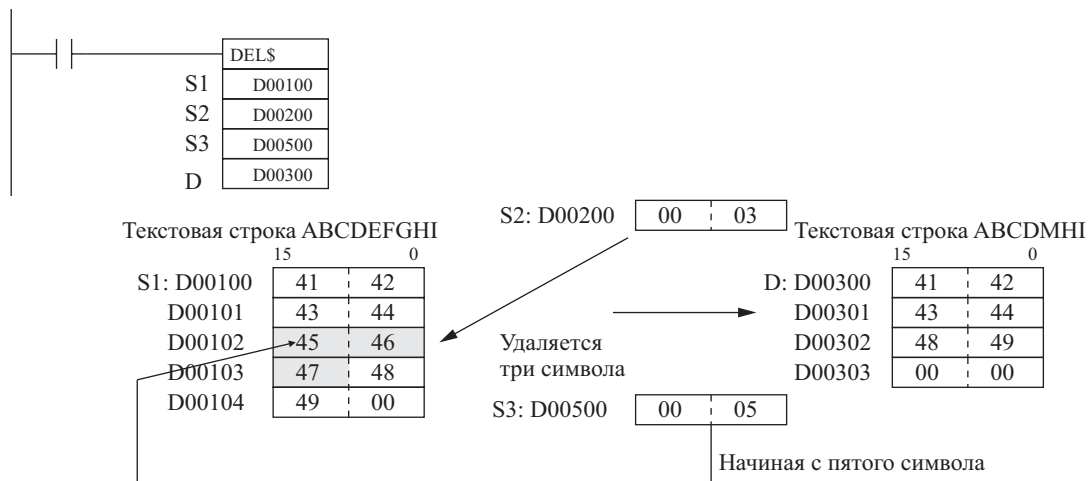
Если количество слов, указанное в S1, превышает величину текстовой строки, указанной в S1, флаг ошибки переводится в состояние ON.

Если количество удаляемых символов (S2) превышает длину текстовой строки S1, все символы текстовой строки (от начала до конца) удаляются. Если задается удаление всех символов текстовой строки S1 (от начала до конца) в D заносится значение 000 (шестн.).

**Пример**

В данном примере команда DEL\$(658) используется для чтения трех символов.



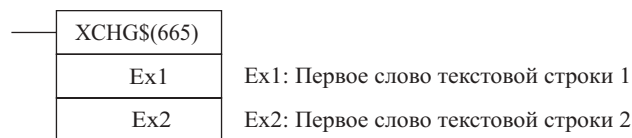


### 3-31-11 Команда обмена содержания текстовых строк EXCHANGE STRING: XCHG\$(665)

#### Назначение

С помощью данной команды осуществляется обмен содержания двух текстовых строк.

#### Символ релейно-контактной схемы



#### Модификации

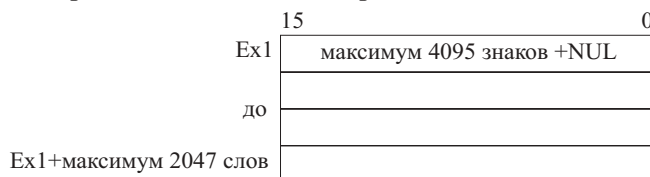
Модификации	Выполняется в каждом цикле для состояния ON	XCHG\$(665)
	Выполняется один раз для дифференцирования вверх	@ XCHG\$(665)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

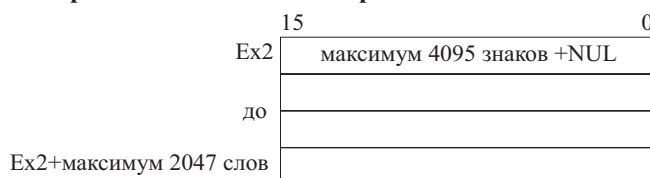
Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

##### Ex1: Первое слово текстовой строки 1



##### Ex2: Первое слово текстовой строки 2



**Примечание:** 1. Данные, содержащиеся в словах Ex1...Ex1+максимальное количество слов (2047) и Ex2...Ex2+максимальное количество слов (2047) должны находиться в одной области данных.

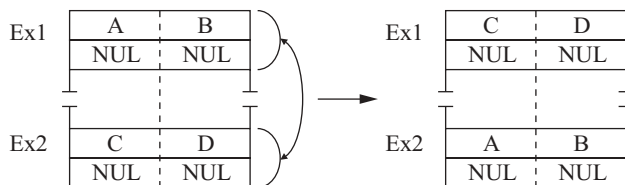
2. Данные от Ex1 до Ex1 + максимальное количество слов (2047) и от Ex2 до Ex2 + максимальное количество слов (2047) перекрываться не могут.

### Спецификации операндов

Область	Ex1	Ex2
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A000...A447 A448...A959	
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_327/67 (n=0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)	
Константы	—	
Регистры данных	—	
Индексные регистры	—	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048 +2047,IR0...-2048...+2047,IR15 DR0...DR15, IR0...IR15 ,IR0(++)...,IR15(++) ,(- -)IR0...,(- -)IR15	

### Описание

По команде XCHG(665) производится замена текстовой строки, указанной в Ex1, другой текстовой строкой, указанной в Ex2. Если одна из текстовых строк Ex1 или Ex2 содержит нули, во вторую строку заносится два символа нуля (0000 шестн.)



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда словом Ex1 или Ex2 указывается более 4095 символов. Переводится в состояние ON, когда данные Ex1 и Ex2 перекрываются. OFF в других случаях.

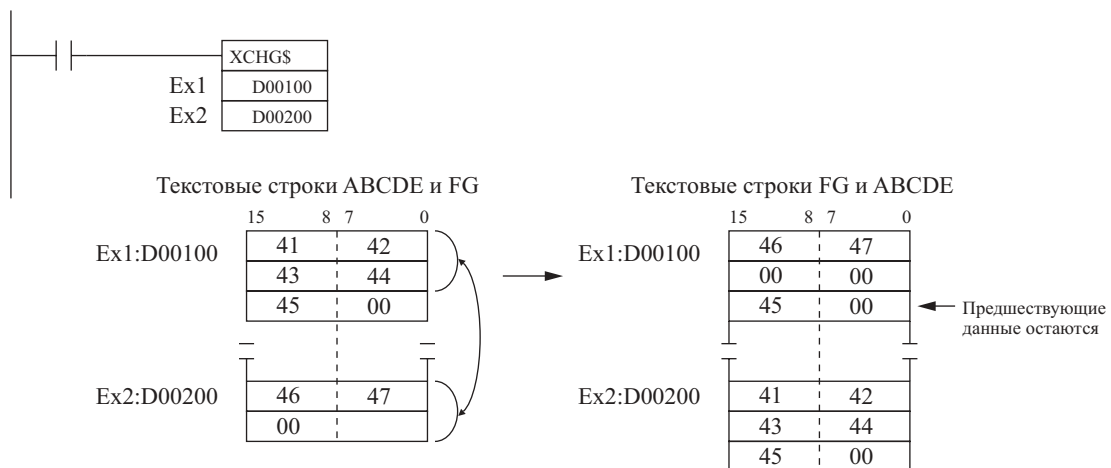
### Меры предосторожности

Максимальное количество символов для Ex1 и Ex2 – 4095 (0FFF шестн.). Если задаваемые значения выйдут за указанный предел, определяется ошибка и флаг ошибки переводится в состояние ON.

Если данные текстовых строк, указанных в Ex1 и Ex2 перекрываются, определяется ошибка и флаг ошибки переводится в состояние ON.

### Пример

В данном примере команда XCHG\$(665) используется для обмена содержанием двух строк.



### 3-31-12 Команда очистки содержания текстовой строки CLEAR STRING: CLR\$(666)

#### Назначение

С помощью данной команды осуществляется очистка содержания текстовой строки

#### Символ релейно-контактной схемы



#### Модификации

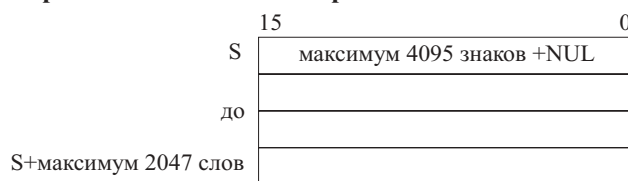
Модификации	Выполняется в каждом цикле для состояния ON	CLR\$(666)
	Выполняется один раз для дифференцирования вверх	@ CLR\$(666)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

#### Операнды

**S:** Первое слово текстовой строки



**Примечание:** 1. Данные, содержащиеся в словах S...S+максимальное количество слов (2047) должны находиться в одной области данных.

#### Спецификации операндов

Область	S
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143
Рабочая область	W000...W511
Область удержания бита (Holding bit area)	H000...H511
Область вспомогательных битов	A448...A959
Область таймера	T0000...T4095
Область счетчика	C0000...C4095
Область DM	D00000...D32767
Область EM, не содержащая банков	E00000...E32767

Область	S
Область EM, содержащая банки	En_00000...En_327/67 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)
Константы	—
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048 +2047,IR0...-2048...+2047,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,(- -)IR0...(- -)IR15

**Описание**

По команде CLR\$(666) производится очистка содержания текстовой строки (т.е. заполнение текстовой строки символом нуля (00 шестн.)), начиная с первого слова, указанного в S до кода окончания 00 (шестн.). Максимальное количество символов, которое может обнуляться с использованием данной команды, равно 4095. Если после 4095 символов отсутствует код окончания 00, обнуляется только 4096 символов.

**Флаги**

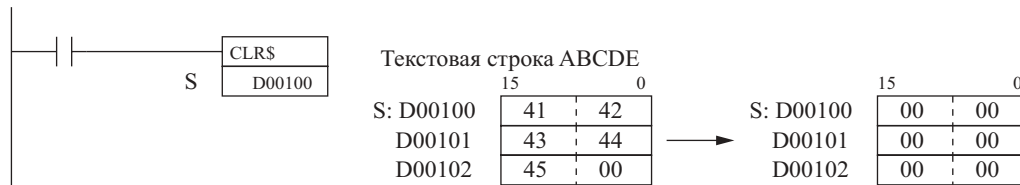
Наименование	Обозначение	Функционирование
Флаг ошибки	ER	OFF

**Меры предосторожности**

После выполнения команды Флаг ошибки переводится в состояние OFF.

**Пример**

В данном примере команда CLR\$(666) используется для очистки содержания строки ABCDE.

**3-31-13 Команда ввода текстовой строки INSERT INTO STRING: IN\$(657)****Назначение**

С помощью данной команды осуществляется ввод указанной текстовой строки в другую текстовую строку.

**Символ релейно-контактной схемы**

IN\$(657)	
S1	S1: Первое слово базовой текстовой строки
S2	S2: Первое слово вводимой текстовой строки
S3	S3: Начальная позиция
D	D: Первое слово назначения

**Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	INSS(657)
	Выполняется один раз для дифференцирования вверх	@ INSS(657)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****S1: Базовая текстовая строка**

	15	0
S1	максимум 4095 знаков +NUL	
до		
S1+максимум 2047 слов		

**S2: Вводимая текстовая строка**

	15	0
S2	максимум 4095 знаков +NUL	
до		
S2+максимум 2047 слов		

**S3: Начальная позиция (0000 0FFF или &0 4095)****D: Первое слово назначения**

	15	0
D	максимум 4095 знаков +NUL	
до		
D+максимум 2047 слов		

- Примечание:**
1. Данные, содержащиеся в словах S1...S1+максимальное количество слов (2047), S2...S2+максимальное количество слов (2047) и D...D+максимальное количество слов (2047), должны находиться в одной области данных.
  2. Данные от S2 до S2+максимальное количество слов (2047) и от D до D+максимальное количество слов (2047) перекрываться не могут. Данные от S1 до S1+максимальное количество слов (2047) и от D до D+максимальное количество слов (2047) могут перекрываться. Данные от S1 до S1+максимальное количество слов (2047) и от S2 до S2+максимальное количество слов (2047) также могут перекрываться.

**Спецификации операндов**

Область	S1	S2	S3	D
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143			
Рабочая область	W000...W511			
Область удержания бита (Holding bit area)	H000...H511			
Область вспомогательных битов	A000...A447 A448...A959			A448...A959
Область таймера	T0000...T4095			
Область счетчика	C0000...C4095			
Область DM	D00000...D32767			
Область EM, не содержащая банков	E00000...E32767			

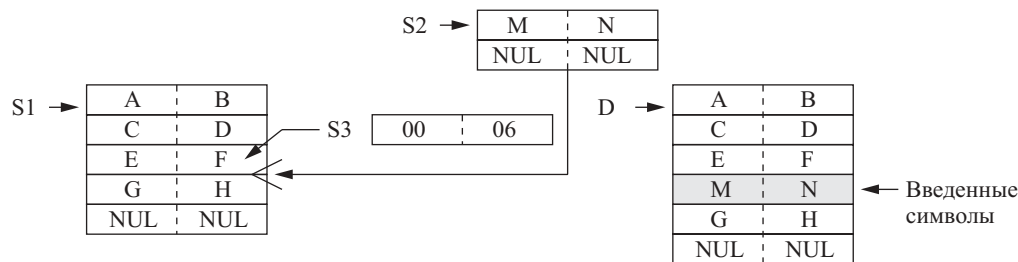
Область	S1	S2	S3	D
Область EM, содержащая банки	En_00000...En_327/67 (n=0...C)			
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)			
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)			
Константы	–	#0000...#FFFF (двоичные) или &0...&4095	–	–
Регистры данных	–	DR0...DR15	–	–
Индексные регистры	–			
Косвенная адресация с использованием индексных регистров	,IR0...IR15 -2048+2047,IR0...-2048...+2047,IR15 DR0...DR15, IR0...IR15 ,IR0+(++)...IR15+(++) ,-(-)IR0...,-(-)IR15			

### Описание

По команде INSS\$(657) производится ввод указанной в S2 текстовой строки в строку, указанную в S1. Начальное слово указывается в S3. Результат заносится в D в виде текстовой строки (в конце данных прибавляется 00).

Максимальное количество вводимых символов – 4095 (0FFF шестн.). Если задаваемое значение выходит за указанный предел, в D заносится только 4095 символов (дополнительно добавляется символ нуля).

Если одна из строк S1 или S2 равна нулю, в D заносится строка S2 без изменения. Если обе строки S1 и S2 равны нулю, в D заносится два символа нуля (0000 шестн.).



### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда словом S1 указывается более 4095 символов. Переводится в состояние ON, когда S3 превышает значение 4095 (0FFF шестн.). Переводится в состояние ON, когда S3 больше S1. OFF в других случаях.
Флаг равенства	=	Переводится в состояние ON, когда в D передается 0000 (шестн.). OFF в других случаях.

### Меры предосторожности

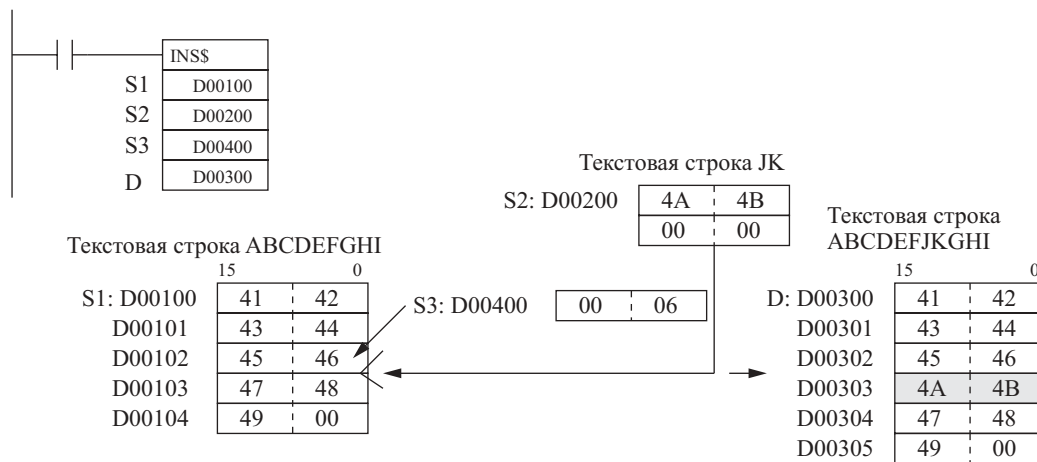
Максимальное количество символов для S1 и S2 – 4095 (0FFF шестн.). Если задаваемое значение выходит за указанный предел (т.е. после 4095 символов отсутствует символ нуля), определяется ошибка и флаг ошибки переводится в состояние ON.

Диапазон задания начальной позиции в S3 – от 0 до 4095. Если значение установки выходит за указанные пределы, определяется ошибка и флаг ошибки переводится в состояние ON.

Когда в D заносится нулевое значение (0000 шестн.) флаг равенства переводится в состояние ON. Не допускайте перекрытия слов назначения, указанных в D, словами текстовой строки S2, т.к. в этом случае команда не может выполняться надлежащим образом.

### Пример

В данном примере команда INSS\$(657) используется для ввода двух символов.



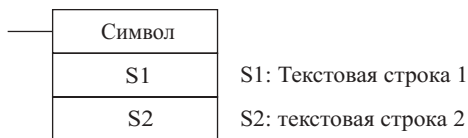
### 3-31-14 Команды сравнения строк (670...675)

#### Назначение

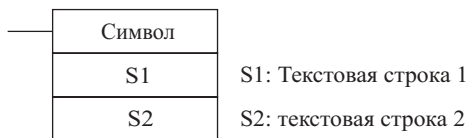
С помощью команд сравнения строк (= \$, < \$, > \$, <= \$, >= \$) осуществляется сравнение содержания двух текстовых строк в ASCII кодах. Если результат сравнения имеет значение истины, создается условие выполнения для команд LOAD, AND, OR.

#### Символ релейно-контактной схемы

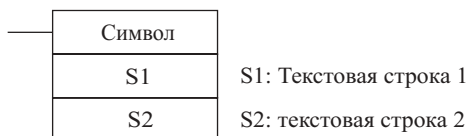
##### LD (Загрузка)



##### AND (Последовательное соединение)



##### OR (Параллельное соединение)

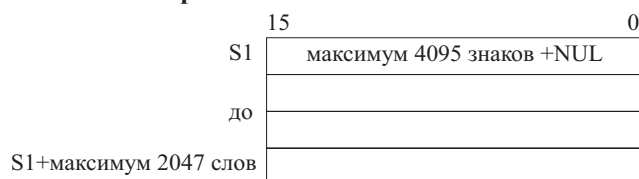
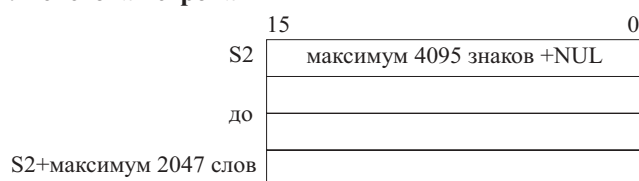


#### Модификации

Модификации	Создает состояние ON в каждом из циклов, когда результатом сравнения является истина.	Команды сравнения текстовых строк.
Указание немедленной регенерации	Не поддерживается	

#### Применяемые области программы

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Да

**Операнды****S1: Текстовая строка 1****S2: Текстовая строка 2**

- Примечание:**
1. Данные, содержащиеся в словах S1...S1+максимальное количество слов (2047) и S2...S2+максимальное количество слов (2047) должны находиться в одной области данных.
  2. Данные от S1 до S1+максимальное количество слов (2047) и от S2 до S2+максимальное количество слов (2047) перекрываться не могут.

**Спецификации операндов**

Область	S1	S2
Область ввода/вывода (область CIO)	CIO 0000...CIO 6143	
Рабочая область	W000...W511	
Область удержания бита (Holding bit area)	H000...H511	
Область вспомогательных битов	A000...A447 A448...A959	
Область таймера	T0000...T4095	
Область счетчика	C0000...C4095	
Область DM	D00000...D32767	
Область EM, не содержащая банков	E00000...E32767	
Область EM, содержащая банки	En_00000...En_32767 (n=0...C)	
Адреса DM/EM памяти при косвенном указании в двоичном коде	@D00000...@D32767 @E00000...@E32767 @En_00000...@En_32767 (n=0...C)	
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	*D00000...*D32767 *E00000...*E32767 *En_00000...*En_32767 (n=0...C)	
Константы	-	
Регистры данных	-	
Индексные регистры	-	
Косвенная адресация с использованием индексных регистров	,IR0...,IR15 -2048+2047,IR0...-2048...+2047,IR15 DR0...DR15,IR0...IR15 ,IR0+(++)...,IR15+(++) ,-(-)IR0...,-(-)IR15	

**Описание**

Команды сравнения текстовых строк осуществляют сравнение содержания текстовых строк, указанных в S1 и S2. Если результат сравнения истинен, в релейно-контактной схеме создается условие выполнения. Максимальное количество символов, содержащихся в S1 и S2 – 4095 (0FFF шестн.).

Команды сравнения строк выражаются с использованием 18-ти мнемонических обозначений, приводимых ниже. (Команды LD, AND, OR в релейно-контактной схеме не используются.)

LD=\$, AND=\$, OR=\$

LD<>\$, AND<>\$, OR<>\$

LD<\$, AND<\$, OR<\$

LD<=\$, AND<=\$, OR<=\$

LD>\$, AND>\$, OR>\$



LD>=\$, AND>=\$, OR>=\$

Следующая ниже таблица содержит подробности, касающиеся функционирования этих команд.

Мнемоническое обозначение (включая функциональный код)	Наименование	Функция
LD=\$ (670)	LOAD STRING EQUALS	Истина, когда текстовая строка S1 равна текстовой строке S2.
AND=\$ (670)	AND STRING EQUALS	
OR=\$ (670)	OR STRING EQUALS	
LD<>\$ (671)	LOAD STRING NOT EQUAL	Истина, когда текстовая строка S1 не равна текстовой строке S2.
AND<>\$ (671)	AND STRING NOT EQUAL	
OR<>\$ (671)	OR STRING NOT EQUAL	
LD<\$ (672)	LOAD STRING LESS THAN	Истина, когда текстовая строка S1 меньше текстовой строки S2.
AND<\$ (672)	AND STRING LESS THAN	
OR<\$ (672)	OR STRING LESS THAN	
LD<=\$ (673)	LOAD STRING LESS THAN OR EQUALS	Истина, когда текстовая строка S1 меньше или равна текстовой строке S2.
AND<=\$ (673)	AND STRING LESS THAN OR EQUALS	
OR<=\$ (673)	OR STRING LESS THAN OR EQUALS	
LD>\$ (674)	LOAD STRING GREATER THAN	Истина, когда текстовая строка S1 больше текстовой строки S2.
AND>\$ (674)	AND STRING GREATER THAN	
OR>\$ (674)	OR STRING GREATER THAN	
LD>=\$ (675)	LOAD STRING LESS THAN OR EQUALS	Истина, когда текстовая строка S1 больше или равна текстовой строке S2.
AND>=\$ (675)	AND STRING GREATER THAN OR EQUALS	
OR>=\$ (675)	OR STRING GREATER THAN OR EQUALS	

### Методы сравнения

При выполнении операций используются следующие ниже методы сравнения:

Первый символ (первый байт) в ASCII коде каждой текстовой строки сравнивается с таким же символом в другой строке. Если два символа ASCII кода не равны, соотношение больше/меньше принимаются истиной для этих текстовых строк. Если два символа ASCII кода равны, производится сравнение следующей пары символов. Если в следующей паре символы не равны, это соотношение принимается в качестве соотношения для обеих строк.

Таким образом, по порядку, символ за символом, производится сравнение двух текстовых строк. Если все символы, включая заключительный символ нуля (00 шестн.) равны, эти строки являются равными строками.

Если две строки имеют разную длину, символы нуля (00 шестн.) добавляются к более короткой строке для заполнения до равной длины, затем сравнение будет выполняться подобно вышеописанной процедуре.

### Примеры выполнения сравнения

AD (414400 шестн.) и BC (424300 шестн.):

AD<BC, т.к. в начале текстовых строк 41 (шестн.) меньше 42 (шестн.).

ADC (41444300 шестн.) и B (4200 шестн.):

ADC<B, т.к. в начале текстовых строк 41 (шестн.) меньше 42 (шестн.).

ABC (41424300 шестн.) и ABD (41424400 шестн.):

ABC<ABD, т.к. в начале текстовых строк символы 41 и 42 соответствуют друг другу, поэтому результат определяется тем, что 43 меньше чем 44.

ABC (41424300 шестн.) и ABD (41424400 шестн.):

ABC>AB, т.к. в начале текстовых строк символы 41 и 42 соответствуют друг другу, поэтому результат определяется тем, что 43 больше чем 00.

AB (414200 шестн.) и AB (414200 шестн.):

AB=AB, т.к. символы 41 и 42 и 00 соответствуют друг другу.

Продолжайте программирование, вводя одну команду после другой, используя команды LD, AND OR в том же порядке. Команды LD и OR могут непосредственно присоединяться к линии шины, а команда AND не может.

**Флаги**

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, когда в S1 или S2 указывается более 4095 символов. OFF в других случаях.
Флаг «Более чем»	>	Переводится в состояние ON, когда в результате сравнения строка S1 больше строки S2. OFF в других случаях
Флаг «Более чем или равно»	>=	Переводится в состояние ON, когда в результате сравнения строка S1 больше или равна строке S2. OFF в других случаях
Флаг равенства	=	Переводится в состояние ON, когда в результате сравнения строка S1 равна строке S2. OFF в других случаях.
Флаг неравенства	<>	Переводится в состояние ON, когда в результате сравнения строка S1 не равна строке S2. OFF в других случаях
Флаг «Менее чем»	<	Переводится в состояние ON, когда в результате сравнения строка S1 меньше строки S2. OFF в других случаях
Флаг «Менее чем или равно»	<=	Переводится в состояние ON, когда в результате сравнения строка S1 меньше или равна строке S2. OFF в других случаях

**Примечание:** Команды сравнения текстовых строк применяются для изменения порядка расстановки текстовых строк в порядке ASCII кода. Например, порядком следования по старшинству в ASCII коде является алфавитный порядок от A до Z, поэтому в результате сравнения строки могут быть расставлены по алфавиту.

**Меры предосторожности**

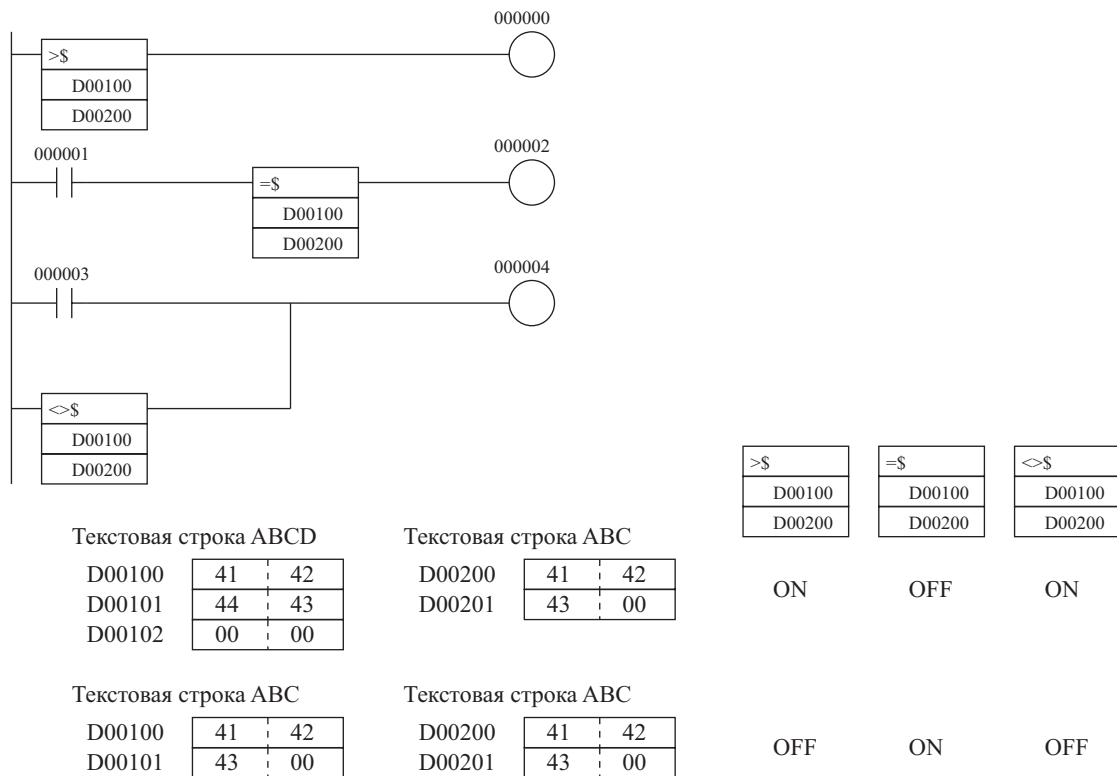
Располагайте правосторонние команды после команд сравнения. Команды сравнения текстовых строк не могут вводиться с правой стороны релейно-контактной схемы.

Данные команды не могут использоваться в последнем звене логического блока.

Максимальное количество символов, которое может подвергаться сравнению, равно 4095 (0FFF шестн.). Если задаваемое значение выходит за указанный предел (т.е. после 4095 символов отсутствует символ нуля), определяется ошибка и флаг ошибки переводится в состояние ON. В этом случае для следующей команды создается условие выполнения OFF.

**Пример**

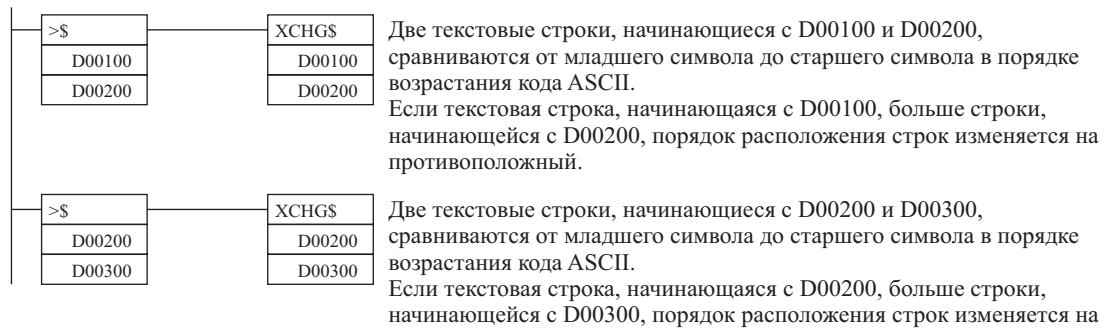
В данном примере команды сравнения текстовой строки используются для сравнения данных



В данном примере три текстовые строки располагаются в алфавитном порядке. Первоначальное расположение является следующим:

D00100: Молоко. D00200: Сок. D00300: Пиво.

При расстановке в алфавитном порядке порядок изменяется следующим образом: пиво, сок, молоко.



## 3-32 Команды управления задачами

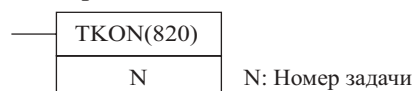
В настоящем разделе приводится описание команд, применяемых для управления задачами.

Команда	Мнемоническое обозначение	Функциональный код	Страница
TASK ON	TKON	820	
TASK OFF	TKOF	821	

## 3-32-1 Команда разрешения выполнения задачи TASK ON: TKON(820)

**Назначение**

Команда разрешает выполнение указанной задачи.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	TKON(820)
	Выполняется один раз для дифференцирования вверх	@ TKON(820)
	Выполняется один раз для дифференцирования вниз.	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Не используется

**Операнды****N: Номер задачи.**

Номер задачи N должен быть постоянной величиной в диапазоне от 00 до 31 в десятичном коде.

**Спецификации операндов**

Область	N
Область ввода/вывода (область CIO)	—
Рабочая область	—
Область удержания бита (Holding bit area)	—
Область вспомогательных битов	—
Область таймера	—
Область счетчика	—
Область DM	—
Область EM, не содержащая банков	—
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	От 00 до 31
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	—

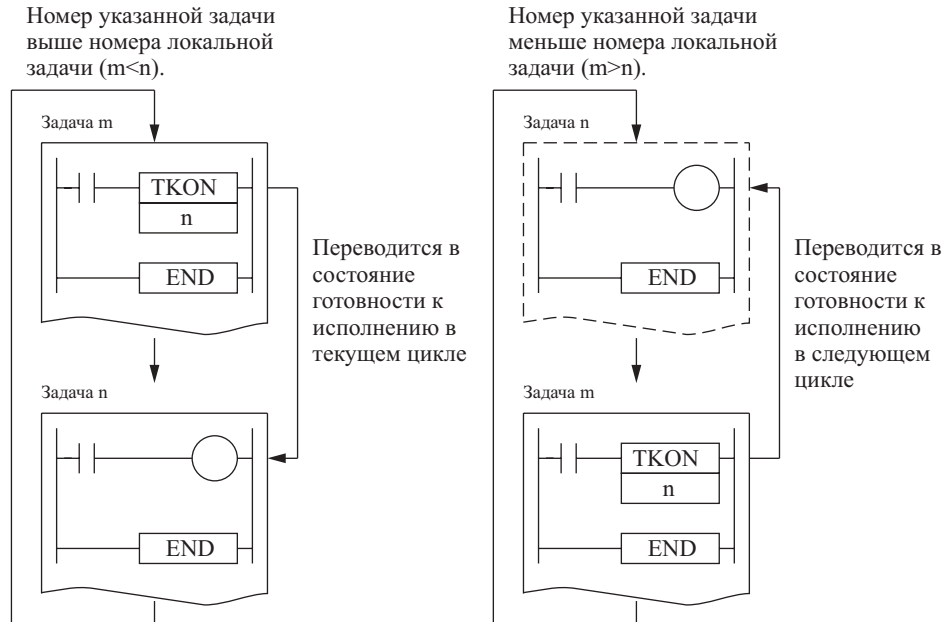
**Описание**

Команда TKON(820) переводит указанную команду в состояние готовности к исполнению и переводит флаг соответствующей задачи (TK00 TK31) в состояние ON.

Данная команда может помещаться только в регулярные (циклические задачи). Она не может вводиться в задачи прерывания.

Задача, указанная в команде ТКОН(820), является готовой к исполнению и в последующих циклах до тех пор, пока она не будет переведена в резерв командой ТКOF(821), периферийным устройством, выполняющим программу СХ- программатор, или FINS командой.

Любая из задач может переводиться в состояние готовности из любой циклической задачи, однако указанная задача не будет выполняться до следующего цикла, если ее номер меньше номера локальной (текущей) задачи. Если номер указываемой задачи больше номера локальной задачи, эта задача будет выполняться в текущем цикле.



Команда ТКОН(820) будет выполняться как команда NOP(000), если указанная задача уже переведена в состояние готовности, или указана текущая задача.

Циклическая задача, которая устанавливается пользователем в состояние готовности при запуске, автоматически переводится в исполняемое состояние при запуске Программируемого контроллера. Все другие циклические задачи переводятся в неисполняемое состояние. Если из Пульта программирования выполняется операция полной очистки, циклическая задача 0 автоматически переводится в состояние готовности к выполнению.

Задача, находящаяся в состоянии готовности, может переводиться в неисполняемое состояние при помощи команды ТКOF(821), с помощью СХ- программатора или командой FINS.

Команда ТКОН(820) может переводить в исполняемое состояние следующие задачи: циклические задачи, которые не переводятся в состояние готовности при запуске Программируемого контроллера, и циклические задачи, которые переведены из состояния готовности в резерв.

Термины «готовые к выполнению» и «выполняемые» не являются равнозначными. Готовые к выполнению задачи выполняются в процессе выполнения циклической программы в порядке их номеров. Задача, готовая к выполнению, не будет выполняться при переводе ее в резерв до того, как программа подойдет к выполнению данной задачи.

### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, если N не является постоянной в диапазоне от 00 до 31. Переводится в состояние ON, когда задача, указанная в N, не существует. Переводится в состояние ON, когда команда ТКОН(820) выполняется в задаче прерывания. OFF в других случаях.
Наименование	Адреса	Функционирование
Флаги задач	TK00 TK31	Данные флаги переводятся в состояние ON, когда соответствующая задача переводится в состояние готовности к выполнению. Флаги переводятся в состояние OFF, когда соответствующая задача является неисполняемой или переведена в резерв. Флаги TK00 TK31 соответствуют номерам задач 00 31.

**Примечание:** Метод ввода номера задачи  $N$  различен при использовании  $SX$ -программатора и Пульта программирования. При использовании  $SX$ -программатора вводите #0 #31, и вводите 00 31 при использовании Пульта программирования.

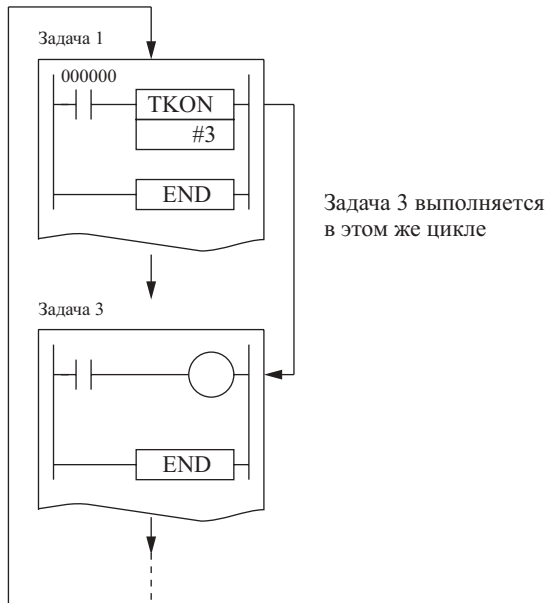
### Меры предосторожности

Данная команда может переводить циклические задачи в состояние готовности к выполнению, однако не может переводить в состояние готовности задачи прерывания.

### Примеры

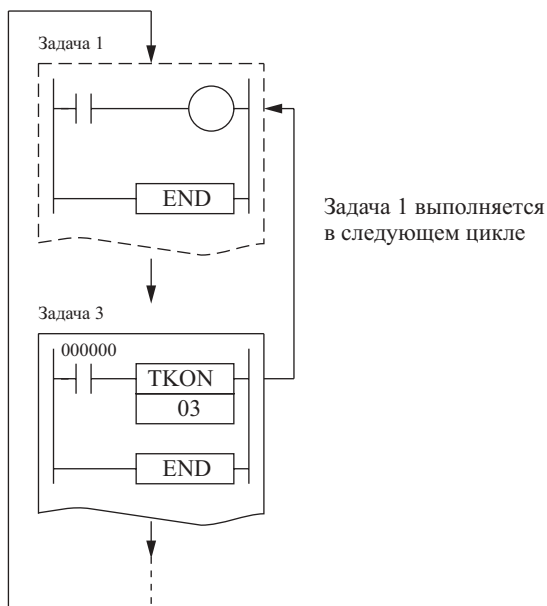
#### Указание более поздней задачи

В следующем ниже примере, когда СЮ 000000 находится в состоянии ON, задача 3 переводится в состояние готовности к выполнению в задаче 1. Задача 3 будет выполняться в текущем цикле, когда программа подойдет к ее выполнению.



#### Указание ранней задачи

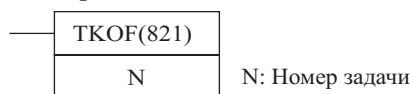
В следующем ниже примере, когда СЮ 000000 находится в состоянии ON, задача 1 переводится в состояние готовности к выполнению в задаче 3. Задача 1 будет выполняться в следующем цикле, когда программа подойдет к ее выполнению.



## 3-32-2 Команда перевода задачи в резерв TASK OFF: TKOF(821)

**Назначение**

Команда переводит указанную задачу в резерв, т.е. делает ее неисполняемой.

**Символ релейно-контактной схемы****Модификации**

Модификации	Выполняется в каждом цикле для состояния ON	TKOF(821)
	Выполняется один раз для дифференцирования вверх	@ TKOF(821)
	Выполняется один раз для дифференцирования вниз	Не поддерживается
Указание немедленной регенерации	Не поддерживается	

**Применяемые области программы**

Области блоков	Области шагов	Подпрограммы	Задачи прерывания
Да	Да	Да	Не используется

**Операнды****N: Номер задачи.**

Номер задачи N должен быть постоянной величиной в диапазоне от 00 до 31 в десятичном коде.

**Спецификации операндов**

Область	N
Область ввода/вывода (область CIO)	—
Рабочая область	—
Область удержания бита (Holding bit area)	—
Область вспомогательных битов	—
Область таймера	—
Область счетчика	—
Область DM	—
Область EM, не содержащая банков	—
Область EM, содержащая банки	—
Адреса DM/EM памяти при косвенном указании в двоичном коде	—
Адреса DM/EM памяти при косвенном указании в двоично-десятичном коде	—
Константы	От 00 до 31
Регистры данных	—
Индексные регистры	—
Косвенная адресация с использованием индексных регистров	—

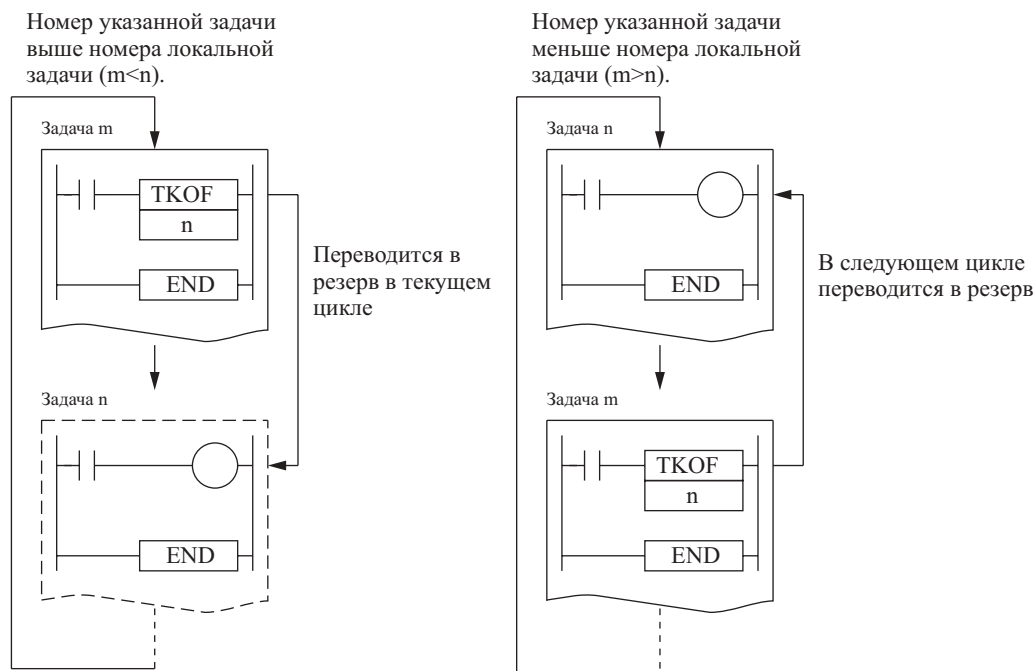
**Описание**

Команда TKOF(821) переводит указанную команду в резерв и переводит флаг соответствующей задачи (TK00 TK31) в состояние OFF.

Задача, указанная в команде TKOF(821), является резервной и в последующих циклах до тех пор, пока она не будет переведена в состояние готовности к выполнению командой TKON(820), периферийным устройством, выполняющим программу CX- программатор, или FINS командой.

Любая из задач может переводиться в резерв из любой другой циклической задачи, однако указанная задача не будет переведена в резерв до следующего цикла, если ее номер меньше номера локальной задачи (она уже была бы выполнена). Если номер указываемой задачи больше номера локальной задачи, эта задача будет переведена в резерв в текущем цикле.

Если в команде TKOF(821) указывается номер локальной задачи, эта задача немедленно переводится в резерв, и ни одна из последующих команд задачи выполняться не будет.



Циклическая задача, которая устанавливается пользователем в состоянии готовности при запуске, автоматически переводится в исполняемое состояние при запуске Программируемого контроллера. Все другие циклические задачи переводятся в неисполняемое состояние.

Задача, находящаяся в состоянии готовности, может переводиться в неисполняемое состояние при помощи команды TKOF(821), с помощью CX- программатора или командой FINS.

Термины «готовые к выполнению» и «выполняемые» не являются равнозначными. Готовые к выполнению задачи будут выполняться в циклической программе по очереди, по порядку номеров этих задач. Задача, готовая к выполнению, не будет выполняться, если она переводится в резерв до того, как программа подойдет к данной задаче.

#### Флаги

Наименование	Обозначение	Функционирование
Флаг ошибки	ER	Переводится в состояние ON, если N не является постоянной в диапазоне от 00 до 31. Переводится в состояние ON, когда задача, указанная в N, не существует. OFF в других случаях.

Наименование	Адреса	Функционирование
Флаги задач	TK00 TK31	Данные флаги переводятся в состояние ON, когда соответствующая задача переводится в состояние готовности к выполнению. Флаги переводятся в состояние OFF, когда соответствующая задача является неисполняемой или переведена в резерв. Флаги TK00 TK31 соответствуют номерам задач 00 31.

**Примечание:** Метод ввода номера задачи N различен при использовании CX- программатора и Пульта программирования. При использовании CX-программатора вводите #0 #31, и вводите 00 31 при использовании Пульта программирования.

#### Меры предосторожности

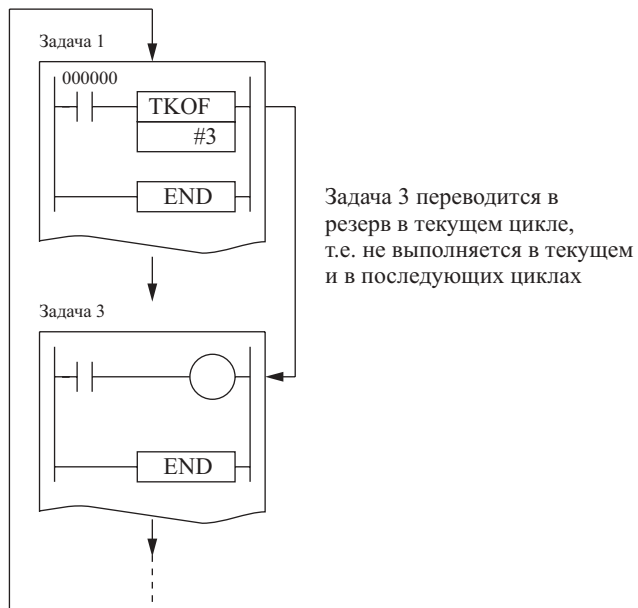
Данная команда может переводить в резерв циклические задачи, однако не может переводить в резерв задачи прерывания.

#### Примеры

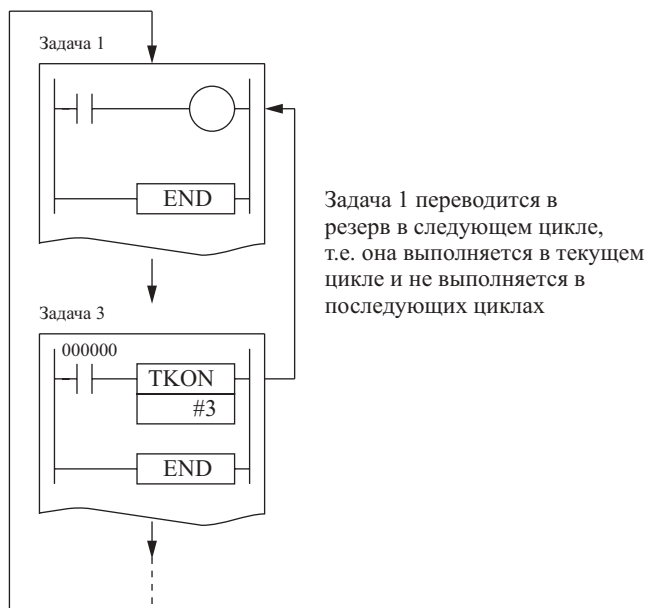
##### Указание более поздней задачи

В следующем ниже примере, когда СЮ 000000 находится в состоянии ON, задача 3 переводится в резерв в задаче 1. Задача 3 не будет выполняться в текущем цикле, когда программа подойдет к ее выполнению.



**Указание ранней задачи**

В следующем ниже примере, когда СЮ 000000 находится в состоянии ON, задача 1 переводится в состояние резерва в задаче 3. Задача 1 не будет выполняться в следующем цикле, когда программа подойдет к ее выполнению.



---

## **Глава 4**

### **Время выполнения команд и количество шагов**

---

*В настоящей главе приводятся данные о времени выполнения команд и количестве шагов для каждой из команд контроллеров серии CSI.*

В следующих ниже таблицах приводится время выполнения команд, используемых в Программируемых контроллерах CS1.

Суммарное время выполнения программы - это время выполнения всех команд в программе пользователя, определенное при вычислении длительности цикла (см. примечание).

**Примечание:** Программы пользователя содержат задачи, которые выполняются в циклических задачах, и задачи прерывания, выполняемые при возникновении условий выполнения.

Время выполнения задач зависит от используемого Модуля центрального процессора (CS1-CPU6\_/CS1-CPU4\_), а также от условий, в которых выполняются эти задачи. Верхняя линия в таблице показывает минимальное время, требуемое для выполнения каждой из команд, и необходимые условия выполнения, нижняя линия – максимальное время и необходимые условия, требуемые для выполнения команды.

Время выполнения может изменяться, когда условие выполнения находится в состоянии OFF.

Таблицы также приводят длину команды в колонке «Длина» (в шагах). Количество шагов в области программы пользователя, требуемых для выполнения каждой из команд, изменяется от 1 до 7 шагов, в зависимости от команды и операнда, используемого командой. Количество шагов в программе не равно количеству команд.

Объем программы для Программируемых контроллеров серии CS1 измеряется в шагах, в отличие от программ для ранее выпускавшихся Программируемых контроллеров, например контроллеров серии C, или CV, объем которых измерялся в словах. В общем случае, один шаг эквивалентен одному слову. Тем не менее, объем памяти, требуемый для некоторых команд контроллеров серии CS1, отличается от указанного, поэтому при преобразовании программ, предназначенных для других контроллеров, в программы для контроллеров серии CS1, расчет объема из соотношения 1 слово= 1 шагу может оказаться ошибочным. Для ознакомления с подробностями преобразования программ для контроллеров предшествующих лет, обратитесь к информации, изложенной в конце главы 4 «Время выполнения команд и количество шагов».

#### Команды для последовательного ввода

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (шаги)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
LOAD	LD	-	1	0.04	0.08	Указание слов	0.04	0.08
				0.50	0.71	Косвенное указание IR+	0.04	0.08
LOAD NOT	LDNOT	-	1	0.04	0.08	Указание слов	0.04	0.08
				0.50	0.71	Косвенное указание IR+	0.04	0.08
AND	AND	-	1	0.04	0.08	Указание слов	0.04	0.08
				0.50	0.71	Косвенное указание IR+	0.04	0.08
AND NOT	AND NOT	-	1	0.04	0.08	Указание слов	0.04	0.08
				0.50	0.71	Косвенное указание IR+	0.04	0.08
OR	OR	-	1	0.04	0.08	Указание слов	0.04	0.08
				0.50	0.71	Косвенное указание IR+	0.04	0.08
OR NOT	OR NOT	-	1	0.04	0.08	Указание слов	0.04	0.08
				0.50	0.71	Косвенное указание IR+	0.04	0.08
AND LOAD	ANDLD	-	1	0.04	0.08	-	0.04	0.08
OR LOAD	ORLD	-	1	0.04	0.08	-	0.04	0.08
NOT	NOT	520	1	0.04	0.08	-	0.04	0.08
CONDITION ON	UP	521	3	0.46	0.54	-	0.12	0.25
CONDITION OFF	DOWN	522	4	0.46	0.54	-	0.12	0.25
LOAD BIT TEST	LDTST	350	4	0.25	0.37	Установка константы в один бит слова	0.21	0.37
				1.21	1.67	Установка 1 бита косвенного IR+ в косвенный IR+	0.21	0.37
LOAD BIT TEST NOT	LD TSTN	351	4	0.25	0.37	Установка константы в один бит слова	0.21	0.37
				1.21	1.67	Установка 1 бита косвенного IR+ в косвенный IR+	0.21	0.37
AND BIT TEST	AND TST	350	4	0.25	0.37	Установка константы в один бит слова	0.21	0.37
				1.21	1.67	Установка 1 бита косвенного IR+ в косвенный IR+	0.21	0.37
AND BIT TEST NOT	AND TSTN	351	4	0.25	0.37	Установка константы в один бит слова	0.21	0.37
				1.21	1.67	Установка 1 бита косвенного IR+ в косвенный IR+	0.21	0.37

Команда	Мнемоника	Код	Длина (шаги)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
OR BIT TEST	OR TST	350	4	0.25	0.37	Установка константы в один бит слова	0.21	0.37
				1.21	1.67	Установка 1 бита косвенного IR+ в косвенный IR+	0.21	0.37
OR BIT TEST NOT	OR TSTN	351	4	0.25	0.37	Установка константы в один бит слова	0.21	0.37
				1.21	1.67	Установка 1 бита косвенного IR+ в косвенный IR+	0.21	0.37

### Команды для последовательного вывода

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (шаги)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
OUTPUT	OUT	–	1	0.17	0.21	Указание слов	0.04	0.08
				0.62	0.83	Косвенное указание IR+	0.04	0.08
OUTPUT NOT	OUT NOT	–	1	0.17	0.21	Указание слов	0.04	0.08
				0.62	0.83	Косвенное указание IR+	0.04	0.08
KEEP	KEEP	011	1	0.25	0.29	Указание слов	0.04	0.08
				0.67	0.87	Косвенное указание IR+	0.04	0.08
DIFFERENTIATE UP	DIFU	013	2	0.46	0.54	Указание слов	0.08	0.17
				0.87	1.12	Косвенное указание IR+	0.08	0.17
DIFFERENTIATE DOWN	DIFD	014	2	0.46	0.54	Указание слов	0.08	0.17
				0.87	1.12	Косвенное указание IR+	0.08	0.17
SET	SET	–	1	0.17	0.21	Указание слов	0.04	0.08
				0.58	0.79	Косвенное указание IR+	0.04	0.08
RESET	RSET	–	1	0.17	0.21	Указание слов	0.04	0.08
				0.58	0.79	Косвенное указание IR+	0.04	0.08
MULTIPLE BIT SET	SETA	530	4	7.8	7.8	С установкой 1 бита	0.21	0.37
				38.8	38.8	С установкой 1000 битов		
MULTIPLE BIT RESET	RSTA	531	4	7.8	7.8	С переустановкой 1 бита	0.21	0.37
				38.8	38.8	С переустановкой 1000 битов		

### Команды для последовательного управления

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
END	END	001	1	4.0	4.0	–	4.0	4.0
NO OPERATION	NOP	000	1	0.08	0.12	–	0.08	0.12
INTERLOCK	IL	002	1	0.12	0.12	–	0.08	0.12
INTERLOCK CLEAR	ILC	003	1	0.12	0.12	–	0.08	0.12
JUMP	JMP	004	2	8.1	8.1	–	4.8	4.8
JUMP END	JME	005	2	3.8	3.8	Когда условия JMP не удовлетворяются	3.8	3.8
CONDITIONAL JUMP	CJP	510	2	7.4	7.4	Когда условия JMP удовлетворяются	5.1	5.1
CONDITIONAL JUMP NOT	CJPN	511	2	8.5	8.5	Когда условия JMP удовлетворяются	4.2	4.2
MULTIPLE JUMP	JMPO	515	1	0.12	0.12	–	0.08	0.12
MULTIPLE JUMP END	JMEO	516	1	0.12	0.12	–	0.08	0.12
FOR LOOP	FOR	512	2	0.12	0.21	Указание константы	0.12	0.21
				0.62	0.83	Косвенное указание IR+	0.12	0.21
BREAK LOOP	BREAK	514	1	0.12	0.12	–	0.08	0.12

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
NEXT LOOP	NEXT	513	1	0.17	0.17	Когда цикл продолжается	0.08	0.12
				0.12	0.12	Когда цикл завершается	0.08	0.12

### Команды управление таймером и счетчиком

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
TIMER	TIM	–	3	0.37	0.42	Константа для заданного значения	0.17	0.29
				0.87	1.12	Косвенный регистр IR+ для заданного значения	0.17	0.29
COUNTER	CNT	–	3	0.37	0.42	Константа для заданного значения	0.17	0.29
				0.87	1.12	Косвенный регистр IR+ для заданного значения	0.17	0.29
HIGH-SPEED TIMER	TIMH	015	3	0.37	0.42	Константа для заданного значения	0.17	0.29
				0.87	1.12	Косвенный регистр IR+ для заданного значения	0.17	0.29
ONE-мс TIMER	TMH	540	3	0.37	0.42	Константа для заданного значения	0.17	0.29
				0.87	1.12	Косвенный регистр IR+ для заданного значения	0.17	0.29
ACCUMULATIVE TIMER	TTIM	087	3	21.4	21.4	–	–	–
				14.8	14.8	При переустановке	–	–
				10.7	10.7	При блокировании	–	–
LONG TIMER	TIML	542	4	12.8	12.8	–	7.8	7.8
				7.8	7.8	При блокировании		
MULTI-OUTPUT TIMER	MTIM	543	4	26.0	26.0	–	0.21	0.37
				7.8	7.8	При переустановке		
REVERSIBLE COUNTER	CNTR	012	3	20.9	20.9	Обычно	17.5	17.5
				16.0	16.0	При переустановке		
				5.7	5.7	При блокировании		
RESET TIMER/COUNTER	CNR	545	3	13.9	13.9	При переустановке 1 слова	0.17	0.29
				5.42 мс	5.42 мс	При переустановке 1,00слов		

### Команды сравнения

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
Входные команды сравнения (без знака)	LD, AND, OR+	300(=)	4	0.21	0.37	Сравнение двух констант	0.21	0.37
		305(<>)	4	1.12	1.58	Сравнение двух косвенных адресов IR+	0.21	0.37
		310(<)	4					
		315(<=)	4					
		320(>)	4					
325(>=)	4							
Входные команды сравнения (двойные, без знака)	LD, AND, OR+ L	301(=)	4	0.29	0.54	Сравнение двух констант	0.29	0.54
		306(<>)	4	1.21	1.75	Сравнение двух косвенных адресов IR+	0.21	0.37
		311(<)	4					
		316(<=)	4					
		321(>)	4					
		326(>=)	4					
Входные команды сравнения (со знаком)	LD, AND, OR+ S	302(=)	4	6.5	6.5	–	0.21	0.37
		307(<>)	4					
		312(<)	4					
		317(<=)	4					
		322(>)	4					
		327(>=)	4					

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
Входные команды сравнения (двойные, со знаком)	LD, AND, OR+, =, <, <=, >, >= + SL	303(=)	4	6.5	6.5	–	0.29	0.54
		308(<=)	4					
		313(<)	4					
		318(<=)	4					
		323(>)	4					
		328(>=)	4					
COMPARE	CMP	020	3	0.17	0.29	Сравнение двух констант	0.17	0.29
				1.08	1.50	Сравнение двух косвенных адресов IR+	0.17	0.29
DOUBLE COMPARE	CMPD	060	3	0.25	0.46	Сравнение двух констант	0.25	0.46
				1.17	1.67	Сравнение двух косвенных адресов IR+	0.17	0.29
SIGNED BINARY COMPARE	CPS	114	3	6.5	6.5	–	0.17	0.29
DOUBLE SIGNED BINARY COMPARE	CPSD	115	3	6.5	6.5	–	0.17	0.29
TABLE COMPARE	TCMP	085	4	21.92	21.92	–	0.21	0.37
MULTIPLE COMPARE	MCMP	019	4	31.2	31.2	–	0.21	0.37
UNSIGNED BLOCK COMPARE	BCMP	068	4	32.6	32.6	–	0.21	0.37

### Команды для перемещения данных

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
MOVE	MOV	021	3	0.25	0.29	Передача константы в слово	0.17	0.29
				1.21	1.62	Передача косвенного IR+ в косвенный IR+	0.17	0.29
DOUBLE MOVE	MOVL	498	3	0.42	0.50	Передача константы в слово	0.21	0.37
				1.42	1.92	Передача косвенного IR+ в косвенный IR+	0.17	0.29
MOVE NOT	MVN	022	3	0.25	0.29	Передача константы в слово	0.17	0.29
				1.21	1.62	Передача косвенного IR+ в косвенный IR+	0.17	0.29
DOUBLE MOVE NOT	MVNL	499	3	0.42	0.50	Передача константы в слово	0.21	0.37
				1.42	1.92	Передача косвенного IR+ в косвенный IR+	0.17	0.29
MOVE BIT	MOV B	082	4	7.5	7.5	–	0.21	0.37
MOVE DIGIT	MOV D	083	4	7.3	7.3	–	0.21	0.37
MULTIPLE BIT TRANSFER	XFR B	062	4	13.6	13.6	Передача 1 бита	0.21	0.37
				269.2	269.2	Передача 255 битов		
BLOCK TRANSFER	XFER	070	4	11.2	11.2	Передача 1 слова	0.21	0.37
				633.5	633.5	Передача 1000 слов		
BLOCK SET	BSET	071	4	8.5	8.5	Задание 1 слова	0.21	0.37
				278.3	278.3	Задание 1000 слов		
DATA EXCHANGE	XCHG	073	3	0.50	0.67	Слово к слову	0.17	0.29
				1.42	1.92	Косвенный IR+ в косвенный IR+	0.17	0.29
DOUBLE DATA EXCHANGE	XCGL	562	3	0.92	1.25	Слово к слову	0.17	0.29
				1.83	2.50	Косвенный IR+ в косвенный IR+	0.17	0.29
SINGLE СЛОВО DISTRIBUTE	DIST	080	4	7.0	7.0	–	0.21	0.37
DATA COLLECT	COLL	081	4	7.1	7.1	–	0.21	0.37
MOVE TO REGISTER	MOV R	560	3	0.42	0.50	Слово в IR	0.21	0.37
				0.96	1.29	Косвенный IR+ в IR	0.17	0.29

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
MOVE TIMER/COUNTER PV TO REGISTER	MOV RW	561	3	0.42	0.50	Слово в IR	0.21	0.37
				0.96	1.29	Косвенный IR+ в IR	0.17	0.29

### Команды смещения данных

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF, мкс.	
				CPU6	CPU4		CPU6	CPU4
SHIFT REGISTER	SFT	010	3	–	–	Переустановка		
				10.4	10.4	Смещение 1 слова	12.7	12.7
				763.1	763.1	Смещение 1000 слов	365.5	365.5
REVERSIBLE SHIFT REGISTER	SFTR	084	4	9.6	9.6	Смещение 1 слова	0.21	0.37
				859.6	859.6	Смещение 1000 слов		
ASYNCHRONOUS SHIFT REGISTER	ASFT	017	4	7.7	7.7	Смещение 1 слова	0.21	0.37
				2.01 мс	2.01 мс	Смещение 1000 слов		
WORD SHIFT	WSFT	016	4	7.8	7.8	Смещение 1 слова	0.21	0.37
				781.7	781.7	Смещение 1000 слов		
ARITHMETIC SHIFT LEFT	ASL	025	2	0.29	0.37	Смещение слов	0.12	0.21
				0.75	1.0	Смещение косвенного IR+	0.12	0.21
DOUBLE SHIFT LEFT	ASLL	570	2	0.50	0.67	Смещение слов	0.12	0.21
				0.96	1.29	Смещение косвенного IR+	0.12	0.21
ARITHMETIC SHIFT RIGHT	ASR	026	2	0.29	0.37	Смещение слов	0.12	0.21
				0.75	1.0	Смещение косвенного IR+	0.12	0.21
DOUBLE SHIFT RIGHT	ASRL	571	2	0.50	0.67	Смещение слов	0.12	0.21
				0.96	1.29	Смещение косвенного IR+	0.12	0.21
ROTATE LEFT	ROL	027	2	0.29	0.37	Вращение слов	0.12	0.21
				0.75	1.0	Вращение косвенного IR+	0.12	0.21
DOUBLE ROTATE LEFT	ROLL	572	2	0.50	0.67	Вращение слов	0.12	0.21
				0.96	1.29	Вращение косвенного IR+	0.12	0.21
ROTATE LEFT WITHOUT CARRY	RLNC	574	2	0.29	0.37	Вращение слов	0.12	0.21
				0.75	1.0	Вращение косвенного IR+	0.12	0.21
DOUBLE ROTATE LEFT WITHOUT CARRY	RLNL	576	2	0.50	0.67	Вращение слов	0.12	0.21
				0.96	1.29	Вращение косвенного IR+	0.12	0.21
ROTATE RIGHT	ROR	028	2	0.29	0.37	Вращение слов	0.12	0.21
				0.75	1.0	Вращение косвенного IR+	0.12	0.21
DOUBLE ROTATE RIGHT	RORL	573	2	0.50	0.67	Вращение слов	0.12	0.21
				0.96	1.29	Вращение косвенного IR+	0.12	0.21
ROTATE RIGHT WITHOUT CARRY	RRNC	575	2	0.29	0.37	Вращение слов	0.12	0.21
				0.75	1.0	Вращение косвенного IR+	0.12	0.21
DOUBLE ROTATE RIGHT WITHOUT CARRY	RRNL	577	2	0.50	0.67	Вращение слов	0.12	0.21
				0.96	1.29	Вращение косвенного IR+	0.12	0.21
ONE DIGIT SHIFT LEFT	SLD	074	3	8.2	8.2	Смещение 1 слова	0.17	0.29
				760.7	760.7	Смещение 1000 слов		
ONE DIGIT SHIFT RIGHT	SRD	075	3	8.7	8.7	Смещение 1 слова	0.17	0.29
				1.07 мс	1.07 мс	Смещение 1000 слов		
SHIFT N-BIT DATA LEFT	NSFL	578	4	10.5	10.5	Смещение 1 бита	0.21	0.37
				55.5	55.5	Смещение 1000 битов		
SHIFT N-BIT DATA RIGHT	NSFR	579	4	10.5	10.5	Смещение 1 бита	0.21	0.37
				69.3	69.3	Смещение 1000 битов		

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF, мкс.	
				CPU6	CPU4		CPU6	CPU4
SHIFT N-BITS LEFT	NASL	580	3	0.29	0.37	Смещение 1 слова на 1 бит	0.17	0.29
				1.25	1.71	Смещение косвенного IR+ на 1 бит	0.17	0.29
DOUBLE SHIFT N-BITS LEFT	NSLL	582	3	0.50	0.67	Смещение 1 слова на 1 бит	0.17	0.29
				1.46	2.0	Смещение косвенного IR+ на 1 бит	0.17	0.29
SHIFT N-BITS RIGHT	NASR	581	3	0.29	0.37	Смещение 1 слова на 1 бит	0.17	0.29
				1.25	1.71	Смещение косвенного IR+ на 1 бит	0.17	0.29
DOUBLE SHIFT N-BITS RIGHT	NSRL	583	3	0.50	0.67	Смещение 1 слова на 1 бит	0.17	0.29
				1.46	2.0	Смещение косвенного IR+ на 1 бит	0.17	0.29

### Команды увеличения/уменьшения

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
INCREMENT BINARY	++	590	2	0.29	0.37	Увеличение слова	0.12	0.21
				0.75	1.0	Увеличение косвенного IR+	0.12	0.21
DOUBLE INCREMENT BINARY	++L	591	2	0.50	0.67	Увеличение слова	0.12	0.21
				0.96	1.29	Увеличение косвенного IR+	0.12	0.21
DECREMENT BINARY	--	592	2	0.29	0.37	Уменьшение слова	0.12	0.21
				0.75	1.0	Уменьшение косвенного IR+	0.12	0.21
DOUBLE DECREMENT BINARY	--L	593	2	0.50	0.67	Уменьшение слова	0.12	0.21
				0.96	1.29	Уменьшение косвенного IR+	0.12	0.21
INCREMENT BCD	++B	594	2	7.4	7.4	–	0.12	0.21
DOUBLE INCREMENT BCD	++BL	595	2	6.1	6.1	–	0.12	0.21
DECREMENT BCD	-B	596	2	7.2	7.2	–	0.12	0.21
DOUBLE DECREMENT BCD	--BL	597	2	7.1	7.1	–	0.12	0.21

### Символьные математические команды

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
SIGNED BINARY ADD WITHOUT CARRY	+	400	4	0.25	0.37	Константа + константа слово	0.21	0.37
				1.71	2.33	Косвенный IR+ + Косвенный IR+ Косвенный IR+	0.21	0.37
DOUBLE SIGNED BINARY ADD WITHOUT CARRY	+L	401	4	0.42	0.54	Константа + константа слово	0.29	0.54
				1.96	2.71	Косвенный IR+ + Косвенный IR+ Косвенный IR+	0.21	0.37
SIGNED BINARY ADD WITH CARRY	+C	402	4	0.25	0.37	Константа + константа слово	0.21	0.37
				1.71	2.33	Косвенный IR+ + Косвенный IR+ Косвенный IR+	0.21	0.37
DOUBLE SIGNED BINARY ADD WITH CARRY	+CL	403	4	0.42	0.54	Константа + константа слово	0.29	0.54
				1.96	2.71	Косвенный IR+ + Косвенный IR+ Косвенный IR+	0.21	0.37
BCD ADD WITHOUT CARRY	+B	404	4	14.0	14.0	–	0.21	0.37
DOUBLE BCD ADD WITHOUT CARRY	+BL	405	4	19.0	19.0	–	0.21	0.37
BCD ADD WITH CARRY	+BC	406	4	14.5	14.5	–	0.21	0.37



Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
DOUBLE BCD ADD WITH CARRY	+BCL	407	4	19.6	19.6	–	0.21	0.37
SIGNED BINARY SUBTRACT WITHOUT CARRY	–	410	4	0.25	0.37	Константа- Константа слово	0.21	0.37
				1.71	2.33	Косвенный IR+ -Косвенный IR+ Косвенный IR+	0.21	0.37
DOUBLE SIGNED BINARY SUBTRACT WITHOUT CARRY	-L	411	4	0.42	0.54	Константа – константа слово	0.29	0.54
				1.96	2.71	Косвенный IR+ -косвенный IR+ косвенный IR+	0.21	0.37
SIGNED BINARY SUBTRACT WITH CARRY	-C	412	4	0.25	0.37	Константа - константа слово	0.21	0.37
				1.71	2.33	Косвенный IR+ -косвенный IR+ косвенный IR+	0.21	0.37
DOUBLE SIGNED BINARY SUBTRACT WITH CARRY	-CL	413	4	0.42	0.54	Константа - константа слово	0.29	0.54
				1.96	2.71	Косвенный IR+ -косвенный IR+ косвенный IR+	0.21	0.37
BCD SUBTRACT WITHOUT CARRY	-B	414	4	13.1	13.1	–	0.21	0.37
DOUBLE BCD SUBTRACT WITHOUT CARRY	-BL	415	4	18.2	18.2	–	0.21	0.37
BCD SUBTRACT WITH CARRY	-BC	416	4	13.8	13.8	–	0.21	0.37
DOUBLE BCD SUBTRACT WITH CARRY	-BCL	417	4	18.8	18.8	–	0.21	0.37
SIGNED BINARY MULTIPLY	*	420	4	0.50	0.58	Константа константа слово	0.21	0.37
				1.96	2.62	Косвенный IR+ косвенный IR+ косвенный IR+	0.21	0.37
DOUBLE SIGNED BINARY MULTIPLY	*L	421	4	11.19	11.19	–	0.21	0.37
UNSIGNED BINARY MULTIPLY	*U	422	4	0.50	0.58	Константа константа – слово	0.21	0.37
				1.96	2.62	Косвенный IR+ косвенный IR+ косвенный IR+	0.21	0.37
DOUBLE UNSIGNED BINARY MULTIPLY	*UL	423	4	10.63	10.63	–	0.21	0.37
BCD MULTIPLY	*B	424	4	12.8	12.8	–	0.21	0.37
DOUBLE BCD MULTIPLY	*BL	425	4	35.2	35.2	–	0.21	0.37
SIGNED BINARY DIVIDE	/	430	4	0.75	0.83	Константа константа слово	0.21	0.37
				2.21	2.87	Косвенный IR+ косвенный IR+ косвенный IR+	0.21	0.37
DOUBLE SIGNED BINARY DIVIDE	/L	431	4	9.8	9.8	–	0.21	0.37
UNSIGNED BINARY DIVIDE	/U	432	4	0.75	0.83	Константа константа слово	0.21	0.37
				2.21	2.87	Косвенный IR+ косвенный IR+ косвенный IR+	0.21	0.37
DOUBLE UNSIGNED BINARY DIVIDE	/UL	433	4	9.1	9.1	–	0.21	0.37
BCD DIVIDE	/B	434	4	15.9	15.9	–	0.21	0.37
DOUBLE BCD DIVIDE	/BL	435	4	26.2	26.2	–	0.21	0.37

## Команды преобразования

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
BCD-TO-BINARY	BIN	023	3	0.25	0.29	В слово после преобразования в константу	0.17	0.29
				1.21	1.62	В слово после преобразования в косвенный IR+	0.17	0.29
DOUBLE BCD-TO-DOUBLE BINARY	BINL	058	3	9.1	9.1	–	0.17	0.29
BINARY-TO-BCD	BCD	024	3	8.3	8.3	–	0.17	0.29
DOUBLE BINARY-TO-DOUBLE BCD	BCDL	059	3	9.2	9.2	–	0.17	0.29
2'S COMPLEMENT	NEG	160	3	0.25	0.29	В слово после преобразования в константу	0.17	0.29
				1.21	1.62	В косвенный IR+ после преобразования в косвенный IR+	0.17	0.29
DOUBLE 2'S COMPLEMENT	NEGL	161	3	0.42	0.5	В слово после преобразования в константу	0.21	0.37
				1.42	1.92	В косвенный IR+ после преобразования в косвенный IR+	0.17	0.29
16-BIT TO 32-BIT SIGNED BINARY	SIGN	600	3	0.42	0.50	В слово после расширения из константы	0.17	0.29
				1.37	1.83	В косвенный IR+ после расширения из косвенного IR+	0.17	0.29
DATA DECODER	MLPX	076	4	8.8	8.8	Декодирование 1 цифры (4 в 16)	0.21	0.37
				12.8	12.8	Декодирование 4 цифр (4 в 16)		
				20.3	20.3	Декодирование 1 цифры (8 в 256)		
				33.4	33.4	Декодирование 2 цифр (8 в 256)		
DATA ENCODER	DMPX	077	4	10.4	10.4	Кодирование 1 цифры (16 в 4)	0.21	0.37
				59.1	59.1	Кодирование 4 цифр (16 в 4)		
				23.6	23.6	Кодирование 1 цифры (256 в 8)		
				92.5	92.5	Кодирование 2 цифр (256 в 8)		
ASCII CONVERT	ASC	086	4	9.7	9.7	Преобразование 1 цифры в ASCII	0.21	0.37
				15.1	15.1	Преобразование 4 цифр в ASCII		
ASCII TO HEX	HEX	162	4	10.1	10.1	Преобразование 1 цифры	0.21	0.37
COLUMN TO LINE	LINE	063	4	29.1	29.1	–	0.21	0.37
LINE TO COLUMN	COLM	064	4	37.3	37.3	–	0.21	0.37
SIGNED BCD-TO-BINARY	BINS	470	4	12.1	12.1	Установка формата данных № 0	0.21	0.37
				12.1	12.1	Установка формата данных №. 1		
				12.7	12.7	Установка формата данных № 2		
				13.0	13.0	Установка формата данных №. 3		
DOUBLE SIGNED BCD-TO-BINARY	BISL	472	4	13.6	13.6	Установка формата данных № 0	0.21	0.37
				13.7	13.7	Установка формата данных № 1		
				14.2	14.2	Установка формата данных № 2		
				14.4	14.4	Установка формата данных № 3		
SIGNED BINARY-TO-BCD	BCDS	471	4	10.6	10.6	Установка формата данных No. 0	0.21	0.37
				10.8	10.8	Установка формата данных No. 1		
				10.9	10.9	Установка формата данных No. 2		
				11.5	11.5	Установка формата данных No. 3		
DOUBLE SIGNED BINARY-TO-BCD	BDSL	473	4	11.6	11.6	Установка формата данных No. 0	0.21	0.37
				11.8	11.8	Установка формата данных No. 1		
				12.0	12.0	Установка формата данных No. 2		
				12.5	12.5	Установка формата данных No. 3		

## Логические команды

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
LOGICAL AND	ANDW	034	4	0.25	0.37	Константа константа слово	0.21	0.37
				1.71	2.33	Косвенный IR+ косвенный IR+ косвенный IR+	0.21	0.37
DOUBLE LOGICAL AND	ANDL	610	4	0.42	0.54	Константа константа слово	0.29	0.54
				1.96	2.71	Косвенный IR+ косвенный IR+ косвенный IR+	0.21	0.37
LOGICAL OR	ORW	035	4	0.25	0.37	Константа V константа слово	0.21	0.37
				1.71	2.33	Косвенный IR+ V косвенный IR+ косвенный IR+	0.21	0.37
DOUBLE LOGICAL OR	ORWL	611	4	0.42	0.54	Константа V константа слово	0.29	0.54
				1.96	2.71	Косвенный IR+ V косвенный IR+ косвенный IR+	0.21	0.37
EXCLUSIVE OR	XORW	036	4	0.25	0.37	Константа константа слово	0.21	0.37
				1.71	2.33	Косвенный IR+ косвенный IR+ косвенный IR+	0.21	0.37
DOUBLE EXCLUSIVE OR	XORL	612	4	0.42	0.54	Константа константа слово	0.29	0.54
				1.96	2.71	Косвенный IR+ косвенный IR+ Косвенный IR+	0.21	0.37
EXCLUSIVE NOR	XNRW	037	4	0.25	0.37	Константа константа слово	0.21	0.37
				1.71	2.33	Косвенный IR+ косвенный IR+ косвенный IR+	0.21	0.37
DOUBLE EXCLUSIVE NOR	XNRL	613	4	0.42	0.54	Константа константа слово	0.29	0.54
				1.96	2.71	Косвенный IR+ косвенный IR+ косвенный IR+	0.21	0.37
COMPLEMENT	COM	029	2	0.29	0.37	Реверсирование слов	0.12	0.21
				0.75	1.0	Реверсирование косвенного IR+	0.12	0.21
DOUBLE COMPLEMENT	COML	614	2	0.50	0.67	Реверсирование слов	0.12	0.21
				0.96	1.29	Реверсирование косвенного IR+	0.12	0.21

## Особые математические команды

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
BINARY ROOT	ROTB	620	3	530.7	530.7	–	0.17	0.29
BCD SQUARE ROOT	ROOT	072	3	514.5	514.5	–	0.17	0.29
ARITHMETIC PROCESS	APR	069	4	32.3	32.3	Определение SIN и COS	0.21	0.37
				78.3	78.3	Определение кусочно-линейной аппроксимации		
FLOATING POINT DIVIDE	FDIV	079	4	176.6	176.6	–	0.21	0.37
BIT COUNTER	BCNT	067	4	22.1	22.1	Вычисление 1 слова	0.21	0.37

## Команды вычислений с плавающей запятой

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
FLOATING 16-BIT	FIX	450	3	14.5	14.5	–	0.17	0.29
FLOATING 32-BIT	FIXL	451	3	14.6	14.6	–	0.17	0.29
16-BIT FLOATING	FLT	452	3	11.1	11.1	–	0.17	0.29

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
32-BIT FLOATING TO	FLTL	453	3	10.8	10.8	–	0.17	0.29
FLOATINGPOINT ADD	+F	454	4	10.2	10.2	–	0.21	0.37
FLOATINGPOINT SUBTRACT	-F	455	4	10.3	10.3	–	0.21	0.37
FLOATINGPOINT DIVIDE	/F	457	4	12.0	12.0	–	0.21	0.37
FLOATINGPOINT MULTIPLY	*F	456	4	10.5	10.5	–	0.21	0.37
DEGREES TO RADIANS	PAD	458	3	14.9	14.9	–	0.17	0.29
RADIANS TO DEGREES	DEG	459	3	14.8	14.8	–	0.17	0.29
SINE	SIN	460	3	61.1	61.1	–	0.17	0.29
COSINE	COS	461	3	44.1	44.1	–	0.17	0.29
TANGENT	TAN	462	3	22.6	22.6	–	0.17	0.29
ARC SINE	ASIN	463	3	24.1	24.1	–	0.17	0.29
ARC COSINE	ACOS	464	3	28.0	28.0	–	0.17	0.29
ARC TANGENT	ATAN	465	3	16.4	16.4	–	0.17	0.29
SQUARE ROOT	SORT	466	3	28.1	28.1	–	0.17	0.29
EXPONENT	EXP	467	3	96.7	96.7	–	0.17	0.29
LOGARITHM	LOG	468	3	17.4	17.4	–	0.17	0.29
EXPONENTIAL POWER	PWR	840	4	181.7	181.7	–	0.21	0.37

#### Команды обработки табличных данных

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
SET STACK	SSET	630	3	8.5	8.5	Указание 5 слов в области стека	0.17	0.29
				276.8	276.8	Указание 1000 слов в области стека		
PUSH ONTO STACK	PUSH	632	3	9.1	9.1	–	0.17	0.29
FIRST IN FIRST OUT	FIFO	633	3	10.6	10.6	Указание 5 слов в области стека	0.17	0.29
				1.13мс	1.13 мс	Указание 1000 слов в области стека		
LAST IN FIRST OUT	LIFO	634	3	9.9	9.9	–	0.17	0.29
DIMENSION RECORD TABLE	DIM	631	5	142.1	142.1	–	0.25	0.46
SET RECORD LOCATION	SETR	635	4	7.0	7.0	–	0.21	0.37
GET RECORD NUMBER	GETR	636	4	11.0	11.0	–	0.21	0.37
DATA SEARCH	SRCH	181	4	19.5	19.5	Поиск 1 слова	0.21	0.37
				3.34 мс	3.34 мс.	Поиск 1000 слов		
SWAP BYTES	SWAP	637	3	13.6	13.6	Перестановка 1 слова	0.17	0.29
				2.82 мс	2.82 мс.	Перестановка 1000 слов		
FIND MAXIMUM	MAX	182	4	24.9	24.9	Поиск 1 слова	0.21	0.37
				3.36 мс	3.36 мс.	Поиск 1000 слов		
FIND MINIMUM	MIN	183	4	25.3	25.3	Поиск 1 слова	0.21	0.37
				3.33 мс	3.33 мс	Поиск 1000 слов		
SUM	SUM	184	4	38.50	38.50	Добавление 1 слова	0.21	0.37
				1.95 мс	1.95 мс	Добавление 1000 слов		
FRAME CHECKSUM	FCS	180	4	28.25	28.25	Для таблицы длиной в 1 слово	0.21	0.37
				2.48 мс	2.48 мс	Для таблицы длиной в 1000 слов		

### Команды управления данными

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
PID CONTROL	PID	190	4	678.2	678.2	Начальное выполнение	15.8	15.8
				474.9	474.9	С выборкой		
				141.3	141.3	Без выборки**		
LIMIT CONTROL	LMT	680	4	22.1	22.1	–	0.21	0.37
DEAD BAND CONTROL	BAND	681	4	22.5	22.5	–	0.21	0.37
DEAD ZONE CONTROL	ZONE	682	4	20.5	20.5	–	0.21	0.37
SCALING	SCL	194	4	56.8	56.8	–	0.21	0.37
SCALING 2	SCL2	486	4	50.7	50.7	–	0.21	0.37
SCALING 3	SCL3	487	4	57.7	57.7	–	0.21	0.37
AVERAGE	AVG	195	4	53.1	53.1	Среднее из одной операции	25.5	25.5
				419.9	419.9	Среднее из 64 операций		

### Команды подпрограмм

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
SUBROUTINE CALL	SBS	091	2	17.0	17.0	–	0.12	0.21
SUBROUTINE ENTRY	SBN	092	2	–	–	–		
SUBROUTINE RETURN	RET	093	1	20.6	20.6	–	20.6	20.6
MACRO	MCRO	099	4	23.3	23.3	–	0.21	0.37

### Команды управления прерываниями

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
SET INTERRUPT MASK	mcKS	690	3	39.5	39.5	–	0.17	0.29
READ INTERRUPT MASK	mcKR	692	3	11.9	11.9	–	0.17	0.29
CLEAR INTERRUPT	CLI	691	3	41.3	41.3	–	0.17	0.29
DISABLE INTERRUPTS	DI	693	1	16.8	16.8	–	0.08	0.12
ENABLE INTERRUPTS	EI	694	1	21.8	21.8	–	21.8	21.8

### Команды управления шагами

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
STEP DEFINE	STEP	008	2	27.1	27.1	Бит управления шагами ON	–	–
				24.4	24.4	Бит управления шагами OFF		
STEP START	SNXT	009	2	10.0	10.0	–	0.12	0.21

### Команды Базового модуля ввода/вывода

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
I/O REFRESH	IORF	097	3	81.7	81.7	Регенерация 1 слова (IN) для Базовых модулей ввода/вывода C200H	0.17	0.29
				86.7	86.7	Регенерация 1 слова (OUT) для Базовых модулей ввода/вывода C200H		
				23.5	23.5	Регенерация 1 слова (IN) для Базовых модулей ввода/вывода CS1		
				25.6	25.6	Регенерация 1 слова (OUT) для Базовых модулей ввода/вывода CS1		
				357.1	357.1	Регенерация 10 слов (IN) для Базовых модулей ввода/вывода C200H		
				407.5	407.5	Регенерация 10 слов (OUT) для Базовых модулей ввода/вывода C200H		
				377.5	377.58	Регенерация 60 слов (IN) для Базовых модулей ввода/вывода CS1		
460.1	460.1	Регенерация 60 слов (OUT) для Базовых модулей ввода/вывода CS1						
7-SEGMENT DECODER	SDEC	078	4	14.1	14.1	–	0.21	0.37
INTELLIGENT I/O READ	IORD	222	4	–	–	–	Длительность чтения/записи зависит от типа Специальных модулей, для которых выполняется операция.	
INTELLIGENT I/O WRITE	IOWR	223	4	–	–	–		

#### Команды последовательного коммуникационного обмена

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
PROTOCOL MACRO	PMCR	260	5	276.8	276.8	Передача 0 слов, прием 0 слов.	0.25	0.46
				305.9	305.9	Передача 249 слов, прием 249 слов.		
TRANSMIT	TXD	236	4	98.8	98.8	Передача 1 байта.	0.21	0.37
				1.10 мс	1.10 мс	Передача 256 байтов.		
RECEIVE	RXD	235	4	131.1	131.1	Прием 1 байта	0.21	0.37
				1.11 мс	1.11 мс	Прием 256 байтов		
CHANGE SERIAL PORT SETUP	STUP	237	3	440.4	440.4	–	0.17	0.29

#### Команды сетевого коммуникационного обмена

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
NETWORK SEND	SEND	090	4	123.9	123.9	–	0.21	0.37
NETWORK RECEIVE	RECV	098	4	124.7	124.7	–	0.21	0.37
DELIVER COMMAND	CMND	490	4	136.8	136.8	–	0.21	0.37

#### Команды для работы с файлами памяти

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
READ DATA FILE	FREAD	700	5	684.1	684.1	Двоичные данные, директория из 2-х символов + имя файла	0.25	0.46
				1.35 мс	1.35 мс	Двоичные данные, директория из 73-х символов + имя файла		

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
				709.8	709.8	Текстовые данные, директория из 2-х символов + имя файла		
				1.37 мс	1.37 мс			
WRITE DATA FILE	FWRIT	701	5	684.7	684.7	Двоичные данные, директория из 2-х символов + имя файла	0.25	0.46
				1.36 мс	1.36 мс	Двоичные данные, директория из 73-х символов + имя файла		
				728.8	728.8	Текстовые данные, директория из 2-х символов + имя файла		
				1.39 мс	1.39 мс	Текстовые данные, директория из 73-х символов + имя файла		

### Команды управления дисплеем

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
DISPLAY MESSAGE	mcG	046	3	14.3	14.3	Вывод сообщения на дисплей	0.17	0.29
				11.3	11.3	Удаление сообщения из дисплея		

### Команды управления часами

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
CALENDAR ADD	CADD	730	4	209.5	209.5	–	0.21	0.37
CALENDAR SUBTRACT	CSUB	731	4	184.1	184.1	–	0.21	0.37
HOURS TO SECONDS	SEC	065	3	35.8	35.8	–	0.17	0.29
SECONDS TO HOURS	HMS	066	3	42.1	42.1	–	0.17	0.29
CLOCK ADJUSTMENT	AD-DATE	735	2	95.9	95.9	–	0.12	0.21

### Команды для выполнения отладки

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
Выборка из памяти отслеживания данных	TRSM	045	1	120.0	120.0	Выборка 1 бита и 0 слов.	6.3	6.3
				1.06 мс	1.06 мс	Выборка 31 бита и 6 слов.		

### Команды выполнения диагностики отказов

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
FAILURE ALARM	FAL	006	3	549.6	549.6	Запись ошибок	0.17	0.29
				244.8	244.8	Удаление ошибок (в порядке приоритета)		
				657.1	657.1	Удаление ошибок (все ошибки)		
				219.4	219.4	Удаление ошибок (индивидуально)		
SEVERE FAILURE ALARM	FALS	007	3	–	–	–	–	–

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max		Время выполнения OFF (мксек.)	
				CPU6	CPU4			CPU6	CPU4
FAILURE POINT DETECTION	FPD	269	4	202.3	202.3	Вывод сообщения не производится	Выполнение	13.5	13.5
				217.6	217.6				
				268.9	268.9	Вывод сообщения	Выполнение		
				283.6	283.6				

### Прочие команды

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max		Время выполнения OFF (мксек.)	
				CPU6	CPU4			CPU6	CPU4
SET CARRY	STC	040	1	0.12	0.12	–	–	0.08	0.12
CLEAR CARRY	CLC	041	1	0.12	0.12	–	–	0.08	0.12
SELECT BANK	EM EMBC	281	2	15.1	15.1	–	–	0.12	0.21
EXTEND MAXIMUM CYCLE TIME	WDT	094	2	19.7	19.7	–	–	0.12	0.21

### Команды программирования блоков

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max		Время выполнения OFF (мксек.)	
				CPU6	CPU4			CPU6	CPU4
BLOCK PROGRAM BEGIN	BPRG	096	2	13.0	13.0	–	–	15.3	15.3
BLOCK PROGRAM END	BEND	801	1	13.1	13.1	–	–	–	–
BLOCK PROGRAM PAUSE	BPPS	811	2	14.9	14.9	–	–	–	–
BLOCK PROGRAM RESTART	BPRS	812	2	8.3	8.3	–	–	–	–
CONDITIONAL BLOCK EXIT	(Условие выполнения) EXIT	806	1	12.9	12.9	Условие EXIT удовлетворяется		–	–
				7.3	7.3	Условие EXIT не удовлетворяется		–	–
CONDITIONAL BLOCK EXIT	EXIT (адрес бита)	806	2	16.3	16.3	Условие EXIT удовлетворяется		–	–
				10.7	10.7	Условие EXIT не удовлетворяется		–	–
CONDITIONAL BLOCK EXIT (NOT)	EXIT NOT (адрес бита)	806	2	16.8	16.8	Условие EXIT удовлетворяется		–	–
				11.2	11.2	Условие EXIT не удовлетворяется		–	–
Ответвление	IF (условие выполнения)	802	1	7.2	7.2	IF действительный		–	–
				10.9	10.9	IF ложный		–	–
Ответвление	IF (номер ретранслятора)	802	2	10.4	10.4	IF действительный		–	–
				14.2	14.2	IF ложный		–	–
Ответвление (NOT)	IF NOT (номер ретранслятора)	802	2	10.9	10.9	IF действительный		–	–
				14.7	14.7	IF ложный		–	–



Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
Ответвление	ELSE	803	1	9.9	9.9	IF действительный	–	–
				11.2	11.2	IF ложный	–	–
Ответвление	IEND	804	1	11.0	11.0	IF действительный	–	–
				7.0	7.0	IF ложный	–	–
ONE CYCLE AND WAIT	WAIT (условие выполнения)	805	1	16.7	16.7	Условие WAIT удовлетворяется	–	–
				6.3	6.3	Условие WAIT не удовлетворяется	–	–
ONE CYCLE AND WAIT	WAIT (номер ретранслятора)	805	2	16.5	16.5	Условие WAIT удовлетворяется	–	–
				9.6	9.6	Условие WAIT не удовлетворяется	–	–
ONE CYCLE AND WAIT (NOT)	WAIT NOT (номер ретранслятора)	805	2	17.0	17.0	Условие WAIT удовлетворяется	–	–
				10.1	10.1	Условие WAIT не удовлетворяется	–	–
COUNTER WAIT	CNTW	814	4	27.4	27.4	Установка по умолчанию	–	–
				28.7	28.7	Обычное выполнение	–	–
HIGHSPEED TIMER WAIT	TMHW	815	3	34.1	34.1	Установка по умолчанию	–	–
				28.9	28.9	Обычное выполнение	–	–
Управление циклом	LOOP	809	1	12.3	12.3		–	–
Управление циклом	LEND (условие выполнения)	810	1	10.9	10.9	Условие LEND удовлетворяется	–	–
				9.8	9.8	Условие LEND не удовлетворяется	–	–
Управление циклом	LEND (номер ретранслятора)	810	2	14.4	14.4	Условие LEND удовлетворяется	–	–
				13.0	13.0	Условие LEND не удовлетворяется	–	–
Управление циклом	LEND NOT (номер ретранслятора)	810	2	14.8	14.8	Условие LEND удовлетворяется	–	–
				13.5	13.5	Условие LEND не удовлетворяется	–	–
TIMER WAIT	TIMW	813	3	33.1	33.1	Установка по умолчанию	–	–
				35.7	35.7	Обычное выполнение	–	–

### Команды обработки текстовых команд

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
MOV STRING	MOV\$	664	3	84.3	84.3	Передача 1 символа	0.17	0.29
				7.27 мс	7.27 мс	Передача 2046 символов		
CONCATENATE STRING	+\$	656	4	167.8	167.8	1 символ + 1 символ	0.21	0.37
				7.42 мс	7.42 мс	2,046 символов + 1 символ		
GET STRING LEFT	LEFT\$	652	4	94.3	94.3	Восстановление 1 символа из 2 символов	0.21	0.37
				7.36 мс	7.36 мс	Восстановление 2046 символов из 2047 символов		
GET STRING RIGHT	RGHT\$	653	4	94.2	94.2	Восстановление 1 символа из 2 символов	0.21	0.37
				11.58 мс	11.58 мс	Восстановление 2046 символа из 2047 символов		
GET STRING MIDDLE	MID\$	654	5	230.2	230.2	Восстановление 1 символа из 3 символов	0.25	0.46
				7.42 мс	7.42 мс	Восстановление 2045 символа из 2047 символов		
FIND IN STRING	FIND\$	660	4	94.1	94.1	Поиск 1 символа из 2 символов	0.21	0.37
				21.95 мс	21.95 мс	Поиск 1024 символа из 2047 символов		
STRING LENGTH	LENS	650	3	33.4	33.4	Удаление 1 символа	0.17	0.29
				4.32 мс	4.32 мс	Обнаружение 2,047 символов		

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
REPLACE IN STRING	RPLC\$	661	6	479.5	479.5	Замена первого символа из 2 символов одним символом	0.29	0.54
				13.46 мс	13.46 мс	Замена первого символа из 2047 символов 1024 символами		
DELETE STRING	DELS\$	658	5	244.6	244.6	Удаление первого символа из 2 символов	0.25	0.46
				11.76 мс	11.76 мс	Удаление первого символа из 2047 символов		
EXCHANGE STRING	XCHGS	665	3	99.0	99.0	Обмен одного символа на другой символ	0.17	0.29
				10.88 мс	10.88 мс	Обмен 2,047 символов на 2,047 других символов		
CLEAR STRING	CLR\$	666	2	37.8	37.8	Очистка 1 символа	0.12	0.32
				5.19 мс	5.19 мс	Очистка 2,047 символов		
INSERT INTO STRING	INSS\$	657	5	428.9	428.9	Ввод 1 символа после первого из двух символов	0.25	0.46
				9.82 мс	9.82 мс	Ввод 1024 символов после первого из 1024 символов		
Команды сравнения строк	LD.AND, OR+, =\$, <>\$, <\$, <=>\$, >\$, >=\$, \$	670 (=)	4	86.2	86.2	Сравнение 1 символа с 1 символом	86.2	86.2
		671 (<>)	4	28.1 мс	28.1 мсек	Сравнение 2047 символов с 2047 символов	28.1 мс	28.1 мс
		672 (<)	4					
		673 (<=)	4					
		674 (>)	4					
		675 (>=)	4					

### Команды управления задачами

При использовании операнда двойной длины, прибавьте 1 к значению, указанному в колонке «Длина».

Команда	Мнемоника	Код	Длина (См. прим)	Время выполнения ON (мксек.)		Условия: Min/Max	Время выполнения OFF (мксек.)	
				CPU6	CPU4		CPU6	CPU4
TASK ON	TKON	820	2	26.3	26.3	–	0.12	0.21
TASK OFF	TKOF	821	2	26.3	26.3	–	0.12	0.21

### Руководство по преобразованию программ, созданных для Программируемых контроллеров OMRON прежних лет выпуска

Руководство по преобразованию программ, созданных для Программируемых контроллеров OMRON прежних лет выпуска (SYSMAC C200HX/HG/HE, CVM1 или серии CV) (объем программы в словах), в программы для Программируемых контроллеров серии CS1 (объем программы в шагах), представлено в следующей ниже таблице.

Прибавьте указанное в таблице значение n к объему программы контроллеров прежних лет выпуска (в словах) для получения объема программы для контроллеров серии CS1 (в шагах).

Шаги CS1 = «а» слов предшествующего Программируемого контроллера + n			
Команды	Изменения	Значение n при преобразовании программы C200HX/HG/HE в программы CS1.	Значение n при преобразовании программы контроллеров серии CV или CVM1 в программы CS1.
Базовые команды	Нет	OUT, SET, RSET, KEEP(011): -1 Другие команды: 0	0
	Дифференцирование вверх	Нет	+1
	Немедленная регенерация	Нет	0
	Дифференцирование вверх и немедленная регенерация	Нет	+2
Специальные команды	Нет	0	-1
	Дифференцирование вверх	+1	0
	Немедленная регенерация	Нет	+3
	Дифференцирование вверх и немедленная регенерация	Нет	+4

---

Например, если команда OUT используется с адресами от CIO 000000 до CIO 25515, объем программы предшествующего Программируемого контроллера будет составлять 2 слова на каждую команду, а для Программируемого контроллера серии CS1 – 1 (2-1) шаг на каждую команду. Например, если используется команда !MOV (команда MOVE с немедленной регенерацией), объем программы предшествующего Программируемого контроллера будет составлять 4 слова на каждую команду, а для Программируемого контроллера серии CS1 – 7 (4+ 3) шагов на каждую команду.

# OMRON

Авторизованный дистрибьютор: