



**Программируемые контроллеры  
серии Alpha**

---

**Руководство по программированию**

### ***Внимание!***

Данное руководство не является официальной документацией OMRON.

В случае возникновения спорных вопросов, связанных с использованием данного Руководства, обращайтесь к официальной документации.

**Программируемые контроллеры  
C200HX/C200HG/C200HE-CPU□□-E  
C200HX/C200HG/C200HE-CPU□□-ZE**

**Руководство по программированию**

**Август 1997**

***Cat. No. W303-E1-1/W322-E1-1 R1.10***



## ***Перед тем, как читать инструкцию:***

Продукция фирмы OMRON создана для использования согласно разрешенным процедурам квалифицированным оператором и только для целей, описанных в данной инструкции.

В данной инструкции приняты следующие обозначения для индикации и классификации предупреждающих сообщений. Обязательно учитывайте эту информацию. Если пренебречь предупреждениями, это может повлечь несчастные случаи с людьми или повреждение оборудования.

**Опасность!** Указывает на информацию, пренебрежение которой с большой долей вероятности повлечет смерть или тяжелые увечья.

**Предупреждение!** Указывает на информацию, пренебрежение которой может повлечь смерть или тяжелые увечья (с меньшей степенью вероятности).

**Внимание!** Указывает на информацию, пренебрежение которой может повлечь относительно серьезные или небольшие травмы, повреждение оборудования или неправильную работу.

## ***Справочная информация об изделиях OMRON***

Все изделия OMRON пишутся в данной инструкции с заглавной буквы. Слово “Модуль” также пишется с заглавной буквы, когда оно относится к продукции OMRON, независимо от того, появляется оно или нет в собственном имени изделия.

Сокращение “Ch”, которое появляется в некоторых режимах индикации и некоторых продуктах OMRON, часто обозначает “слово” (word) и в документации имеет сокращенное обозначение “Wd”.

Сокращение “ПК” (PC) означают Программируемый Контроллер (Programmable Controller) и в других смыслах не используется.

## ***Средства выделения информации***

В левой колонке данной инструкции появляются следующие заголовки для облегчения определения типа информации.

**Замечание** Указывает информацию, заслуживающего особого интереса для эффективной и удобной работы изделий OMRON.

**1, 2, 3,...** Указывает на перечисления того или иного рода, такие, как процедуры, списки и т.д.

### **OMRON, 1996**

Все права сохранены. Никакую часть данного документа нельзя размножить, загружать в информационно-поисковые системы или передавать в любой форме, механической, электрической, фотокопированием, магнитозаписью или какой-либо другой, без письменного разрешения OMRON.

Патентной ответственности за использование информации в данном документе не несет. Более того, поскольку OMRON постоянно старается улучшать свою продукцию, информация в данной инструкции может измениться без уведомления. При подготовке данной инструкции были приняты все меры предосторожности. Однако OMRON не принимает на себя ответственности за ошибки или пропуски. Не принимается также никакая ответственность за ущерб, нанесенный вследствие применения информации, содержащейся в данном документе.



# Содержание

Меры предосторожности . . . . .	15
1. Для кого предназначена данная инструкция . . . . .	16
2. Общие меры предосторожности. . . . .	16
3. Меры предосторожности по безопасности . . . . .	16
4. Требования к окружающей среде . . . . .	16
5. Меры предосторожности при применении . . . . .	17
6. Соответствие директивам ЕС . . . . .	18
1. Введение. . . . .	19
1.1 Общие сведения . . . . .	20
1.2 Происхождение логики ПК . . . . .	20
1.3 Терминология ПК . . . . .	21
1.4 Терминология продукции OMRON . . . . .	21
1.5 Краткое описание работы ПК. . . . .	22
1.6 Периферийные устройства . . . . .	23
1.7 Имеющиеся инструкции. . . . .	23
1.8 Характеристики C200HX/HG/HE . . . . .	25
1.8.1 Усовершенствования в C200HX/HG/HE - ZE . . . . .	25
1.8.2 Возможности C200HS и C200HX/HG/HE . . . . .	26
1.8.3 Совместимость программ . . . . .	28
2. Аппаратная часть . . . . .	31
2.1 Компоненты ЦПУ . . . . .	32
2.1.1 Индикаторы ЦПУ. . . . .	33
2.1.1 Подключение периферийного устройства . . . . .	33
2.2 Конфигурация ПК . . . . .	34
2.3 Характеристики ЦПУ . . . . .	36
2.4 Кассеты памяти . . . . .	36
2.4.1 Установочные параметры и начальные установки аппаратуры . . . . .	37
2.4.2 Запись/чтение содержимого UM . . . . .	38
2.4.3 Запись/чтение содержимого IOM . . . . .	39
2.5 Работа без батареи . . . . .	40
2.6 Переключатель DIP ЦПУ . . . . .	41
3. Области памяти . . . . .	43
3.1 Введение. . . . .	44
3.1.1 Общий обзор области данных . . . . .	44
3.1.2 Общие сведения об области IR/SR. . . . .	45
3.2 Структура области данных . . . . .	46
3.3 Область IR (внутренние реле) . . . . .	49
3.4 Область SR (специальных реле) . . . . .	53
3.4.1 Система SYSMAC NET/SYSMAC LINK . . . . .	59
3.4.2 Системы удаленных входов/выходов. . . . .	61
3.4.3 Флаги и биты управления системы связи. . . . .	61
3.4.4 Бит удержания принудительного состояния . . . . .	63
3.4.5 Бит удержания состояния входов/выходов. . . . .	64
3.4.6 Бит сброса выходов . . . . .	64
3.4.7 Область FAL (предупреждение об ошибке) . . . . .	64
3.4.8 Флаг падения напряжения батареи. . . . .	65
3.4.9 Флаг ошибки времени цикла . . . . .	65
3.4.10 Флаг ошибки сверки входов/выходов . . . . .	65

3.4.11	Флаг первого цикла . . . . .	65
3.4.12	Биты синхронизирующих импульсов . . . . .	65
3.4.15	Флаг ошибки специальных модулей . . . . .	66
3.4.16	Флаг ошибки исполнения команды, ER . . . . .	66
3.4.17	Флаги арифметики . . . . .	66
3.4.18	Области подпрограмм прерывания . . . . .	67
3.4.19	Области связи порта RS-232C . . . . .	67
3.4.20	Области связи периферийного порта . . . . .	68
3.4.21	Области кассеты памяти . . . . .	68
3.4.22	Биты ошибки передачи . . . . .	69
3.4.23	Области памяти программы в виде РКС . . . . .	69
3.4.24	Флаги ошибки памяти . . . . .	70
3.4.25	Флаги сохранения данных . . . . .	70
3.4.26	Флаги ошибки передачи . . . . .	70
3.4.27	Флаги ошибки установочных параметров ПК . . . . .	70
3.4.28	Текущее время и отображение состояния клавиатуры . . . . .	70
3.4.29	Флаги ошибки группы 2 . . . . .	71
3.4.30	Биты перезапуска и флаги ошибок Специальных модулей . . . . .	71
3.5	Область AR (Вспомогательных реле) . . . . .	71
3.5.1	Перезапуск модулей специальных входов/выходов . . . . .	73
3.5.2	Флаги ошибки ведомой панели . . . . .	73
3.5.3	Флаги ошибки группы 2 . . . . .	73
3.5.4	Флаги ошибок модулей оптических входов/выходов и терминалов входов/выходов . . . . .	74
3.5.5	Установочные параметры системы связи SYSTEM LINK . . . . .	74
3.5.6	Биты протокола ошибок . . . . .	75
3.5.7	Флаги активного узла . . . . .	75
3.5.8	Время обслуживания системы SYSMAC LINK/SYSMAC NET . . . . .	76
3.5.9	Области и Биты календарь/часы . . . . .	76
3.5.10	Биты клавиш режима TERMINAL . . . . .	76
3.5.11	Счетчик отключений питания . . . . .	77
3.5.12	SYSMAC LINK - Флаги периферийного устройства . . . . .	77
3.5.13	Флаг времени цикла . . . . .	77
3.5.14	Флаги установленных модулей связи . . . . .	78
3.5.15	Флаг устройства, установленного на ЦПУ . . . . .	78
3.5.16	Бит переключения FPD . . . . .	78
3.5.17	Флаги трассировки данных и биты управления . . . . .	78
3.5.18	Индикаторы времени цикла . . . . .	78
3.6	Область DM (Data Memory) . . . . .	78
3.6.1	Область расширенных DM . . . . .	79
3.6.2	Область модулей специальных входов/выходов . . . . .	80
3.6.3	Специальные Модули . . . . .	80
3.6.4	Область протокола ошибок . . . . .	82
3.6.5	Установочные параметры ПК . . . . .	84
3.6.6	Установочные параметры панели связи . . . . .	89
3.6.7	Установочные параметры области модулей специальных входов/выходов . . . . .	92
3.7	Область HR (Holding Relay) . . . . .	93
3.8	Область TC (таймеров/ счетчиков) . . . . .	93
3.9	Область LR (Link Relay) . . . . .	94
3.10	Область UM . . . . .	94
3.11	Область TR (временные ячейки) . . . . .	95
3.12	Область EM (Расширенная область DM) . . . . .	95
3.12.1	Использование области EM . . . . .	96
3.12.2	Текущий банк EM . . . . .	97
4.	Написание и ввод программ . . . . .	99
4.1	Основной алгоритм . . . . .	100



4.2 Терминология команд . . . . .	100
4.3 Объем памяти программ . . . . .	101
4.4 Базовые релейно-контактные схемы. . . . .	101
4.4.1 Базовые термины . . . . .	102
4.4.2 Мнемокод . . . . .	102
4.4.3 Команды РКС. . . . .	103
4.4.4 Вывод и Вывод инверсии (OUTPUT и OUTPUT NOT) . . . . .	106
4.4.5 Команда END. . . . .	106
4.4.6 Блочные команды. . . . .	107
4.4.7 Кодирование нескольких выходных команд . . . . .	115
4.5 Консоль программирования . . . . .	116
4.5.1 Клавиатура . . . . .	116
4.5.2 Режимы работы ПК . . . . .	118
4.5.3 Переключатель языка сообщений . . . . .	119
4.6 Подготовка к работе. . . . .	119
4.6.1 Ввод пароля . . . . .	120
4.6.2 Зуммер . . . . .	120
4.6.3 Очистка памяти . . . . .	120
4.6.4 Регистрация таблицы входов/выходов . . . . .	123
4.6.5 Очистка сообщений об ошибках . . . . .	124
4.6.6 Сверка таблицы входов/выходов . . . . .	124
4.6.7 Чтение таблицы входов/выходов . . . . .	125
4.6.8 Очистка таблицы входов/выходов . . . . .	128
4.6.9 Передача таблицы связи SYSMAC NET . . . . .	128
4.7 Ввод, изменение и проверка программы . . . . .	129
4.7.1 Установка и чтение из адреса памяти программ. . . . .	130
4.7.2 Ввод и редактирование программ . . . . .	130
4.7.3 Проверка программы . . . . .	133
4.7.4 Индикация времени цикла. . . . .	135
4.7.5 Поиск в программе. . . . .	135
4.7.6 Вставка и удаление команд . . . . .	137
4.7.7 Разветвляющиеся командные линии . . . . .	140
4.7.8 Переходы . . . . .	145
4.8 Управление состоянием битов . . . . .	147
4.8.1 Включить на 1 цикл (DIFFERENTIATE UP и DIFFERENTIATE DOWN) . . . . .	147
4.8.2 Сохранить (KEEP) . . . . .	148
4.8.3 Биты самоподдержки . . . . .	148
4.9 Рабочие биты (внутренние реле) . . . . .	149
4.10 Предосторожности при программировании . . . . .	151
4.11 Исполнение программы . . . . .	153
4.12 Программы работы с модулями специальных входов/выходов . . . . .	153
4.12.1 Перезапуск модулей специальных входов/выходов . . . . .	153
4.12.2 Программа обработки ошибок модулей специальных входов/выходов. . . . .	154
4.12.3 Изменение параметров модулей специальных входов/выходов . . . . .	154
4.12.4 Интервал обновления входов/выходов модуля специальных входов/выходов . . . . .	155
4.12.5 Сокращение времени цикла . . . . .	156
4.13 Программирование модуля аналогового таймера . . . . .	157
4.13.1 Работа . . . . .	158
4.13.2 Распределение битов и установка переключателя DIP . . . . .	158
4.13.3 Пример программы. . . . .	159
5. Набор команд . . . . .	163
5.1 Общие сведения . . . . .	164
5.2 Формат команды . . . . .	164
5.3 Области данных, значения определителей и флаги . . . . .	164

5.4 Команды фронта 0/1 . . . . .	166
5.5 Дополнительные команды . . . . .	167
5.5.1 Дополнительные команды C200HX/HG/HE-CPU__-E . . . . .	167
5.5.2 Дополнительные команды C200HX/HG/HE-CPU__-ZE . . . . .	168
5.6 Кодирование выходных команд (“правосторонних”) . . . . .	170
5.7 Списки команд. . . . .	172
5.7.1 Функциональные коды C200HX/HG/HE-CPU__-E . . . . .	172
5.7.2 Алфавитный список мнемоники C200HX/HG/HE-CPU__-E . . . . .	173
5.7.3 Функциональные коды C200HX/HG/HE-CPU__-ZE. . . . .	177
5.7.4 Алфавитный список мнемоники C200HX/HG/HE-CPU__-ZE . . . . .	177
5.8 Команды релейно-контактной схемы . . . . .	182
5.8.1 LOAD, LOAD NOT, AND, AND NOT, OR и OR NOT. . . . .	182
5.8.2 AND LOAD и OR LOAD . . . . .	183
5.9 Команды управления битами. . . . .	184
5.9.1 OUT и OUT NOT - Вывод и Вывод инверсии . . . . .	184
5.9.2 DIFU и DIFD - Установка бита на 1 цикл (после фронта 0/1 и 1/0) . . . . .	184
5.9.3 SET и RESET (Установка и сброс) . . . . .	186
5.9.4 KEEP - Сохранить состояние . . . . .	188
5.9.5 TST и TSTN - Проверка бита . . . . .	190
5.10 IL и ILC - Секция INTERLOCK и INTERLOCK CLEAR . . . . .	191
5.11 JMP и JME - Переход и конец перехода . . . . .	193
5.12 END - Конец программы . . . . .	195
5.13 NOP - Нет операции . . . . .	195
5.14 Команды таймеров и счетчиков. . . . .	196
5.14.1 TIM - Таймер . . . . .	196
5.14.2 TIMH - Высокоскоростной таймер . . . . .	201
5.14.3 TTIM - Суммирующий таймер . . . . .	202
5.14.4 CNT - Счетчик. . . . .	203
5.14.5 CNTR - Реверсивный счетчик . . . . .	206
5.15 Команды сдвига . . . . .	208
5.15.1 SFT - Сдвиговый регистр. . . . .	208
5.15.2 SFTR - Реверсивный регистр сдвига . . . . .	210
5.15.3 ASL - Арифметический сдвиг влево . . . . .	212
5.15.4 ASR - Арифметический сдвиг вправо . . . . .	212
5.15.5 ROL - Циклический сдвиг влево . . . . .	213
5.15.6 ROR - Циклический сдвиг вправо . . . . .	213
5.15.7 SLD - Сдвиг влево на одну цифру . . . . .	214
5.15.8 SRD - Сдвиг вправо на одну цифру . . . . .	215
5.15.9 WSFT - Сдвиг слова . . . . .	215
5.15.10 ASFT - Асинхронный регистр сдвига . . . . .	216
5.16 Пересылка данных. . . . .	218
5.16.1 MOV - Пересылка. . . . .	218
5.16.2 MVN - Пересылка инверсного значения. . . . .	218
5.16.3 BSET - Заполнение блока . . . . .	219
5.16.4 XFER - Пересылка блока . . . . .	220
5.16.5 XCHG - Обмен данных . . . . .	221
5.16.6 DIST - Распределение одного слова . . . . .	222
5.16.7 COLL - Сбор данных . . . . .	223
5.16.8 MOVБ - Переслать бит . . . . .	226
5.16.9 MOVD - Переслать цифру . . . . .	227
5.16.10 XFRB - Переслать биты. . . . .	228
5.16.11 XFR2 - Пересылка блока EM . . . . .	229
5.16.12 VXF2 - Пересылка банка EM . . . . .	230
5.16.13 VXFR - Пересылка банка EM . . . . .	231

5.17 Команды сравнения . . . . .	233
5.17.1 MCMP - Сравнение нескольких слов . . . . .	233
5.17.2 CMP - Сравнение . . . . .	234
5.17.3 CMPL - Сравнение чисел двойной длины. . . . .	236
5.17.4 BCMP - Блочное сравнение. . . . .	237
5.17.5 TCMP - Табличное сравнение . . . . .	238
5.17.6 ZCP - Сравнение с зоной. . . . .	239
5.17.7 ZCPL - Сравнение с зоной чисел двойной длины . . . . .	240
5.17.8 CPS - Сравнение двоичных чисел со знаком . . . . .	241
5.17.9 CPSL - Сравнение двоичных слов двойной длины со знаком . . . . .	242
5.7.10 Команды входного сравнения . . . . .	242
5.18 Команды преобразования данных . . . . .	245
5.18.1 BIN - Преобразование двоично-десятичного числа в двоичное . . . . .	245
5.18.2 BINL - Преобразование двоично-десятичного числа двойной длины в двоичное двойной длины . . . . .	245
5.18.3 BCD - Преобразование из двоичного вида в двоично-десятичный . . . . .	246
5.18.4 BCDL - Преобразование двоичного числа двойной длины в двоично-десятичное число двойной длины . . . . .	247
5.18.5 SEC - Преобразование Часы-в-секунды . . . . .	247
5.18.6 HMS - Преобразование Секунды-в-часы . . . . .	248
5.18.7 MLPX - Преобразовать 4-в-16 / 8-в-256 . . . . .	249
5.18.8 DMPX - Преобразовать 16-в-4 / 256-в-8 . . . . .	252
5.18.9 SDEC - Преобразование в коды 7-сегментного индикатора . . . . .	255
5.18.10 ASC - Преобразование в коды ASCII. . . . .	257
5.18.11 HEX - Преобразование из ASCII в 16-ричное число . . . . .	259
5.18.12 SCL - Масштабирование . . . . .	262
5.18.13 LINE - Преобразование Столбец-в-строку . . . . .	264
5.18.14 COLM - Преобразование Строка-в-столбец . . . . .	265
5.18.15 NEG - Дополнение до 2 . . . . .	265
5.18.16 NEGL - Дополнение до 2 слова двойной длины. . . . .	266
5.19 Символьные арифметические команды . . . . .	268
5.19.1 Сложение двоичных чисел: +, +L, +C, +CL . . . . .	268
5.19.2 Сложение двоично-десятичных чисел: +B/ +BL/+BC/+BCL . . . . .	270
5.19.3 Вычитание двоичных чисел: -, -L, -C, -CL . . . . .	272
5.19.4 Вычитание двоично-десятичных чисел: -B/ -BL/-BC/-BCL . . . . .	277
5.19.5 Умножение двоичных чисел: *, *L, *U, *UL . . . . .	280
5.19.6 Умножение двоично-десятичных чисел: *B, *BL . . . . .	283
5.19.7 Деление двоичных чисел: /, /L, /U, /UL . . . . .	284
5.19.8 Деление двоично-десятичных чисел: /B, /BL . . . . .	286
5.20 Команды вычислений с двоично-десятичными числами . . . . .	289
5.20.1 INC - Инкремент . . . . .	289
5.20.2 DEC - Декремент двоично-десятичного числа . . . . .	289
5.20.3 STC - Установка флага переноса . . . . .	290
5.20.4 CLC - Сброс флага переноса . . . . .	290
5.20.5 ADD - Сложение двоично-десятичных чисел . . . . .	290
5.20.6 ADDL - Сложение двоично-десятичных чисел двойной длины . . . . .	292
5.20.7 SUB - Вычитание двоично-десятичных чисел . . . . .	293
5.20.8 SUBL - Вычитание двоично-десятичных чисел двойной длины . . . . .	295
5.20.9 MUL - Умножение двоично-десятичных чисел . . . . .	296
5.20.10 MULL - Умножение двоично-десятичных чисел двойной длины . . . . .	297
5.20.11 DIV - Деление двоично-десятичных чисел. . . . .	298
5.20.12 DIVL - Деление двоично-десятичных чисел двойной длины . . . . .	299
5.20.13 FDIV - Деление чисел с плавающей запятой . . . . .	299
5.20.14 ROOT - Квадратный корень. . . . .	302
5.21 Команды вычислений с двоичными числами . . . . .	305
5.21.1 ADB - Сложение двоичных чисел . . . . .	305
5.21.2 SBB - Вычитание двоичных чисел. . . . .	307

5.21.3	MLB - Умножение двоичных чисел . . . . .	310
5.21.4	DVB - Деление двоичных чисел . . . . .	310
5.21.5	ADBL - Сложение двоичных чисел двойной длины. . . . .	311
5.21.6	SBBL - Вычитание двоично-десятичных чисел двойной длины . . . . .	313
5.21.7	MBS - Умножение двоичных чисел со знаком . . . . .	315
5.21.8	MBSL - Умножение двоичных чисел двойной длины . . . . .	316
5.21.9	DBS - Деление двоичных чисел со знаком . . . . .	317
5.21.10	DBSL - Деление двоичных чисел двойной длины со знаком . . . . .	318
<b>5.22</b>	<b>Специальные математические команды . . . . .</b>	<b>319</b>
5.22.1	MAX - Найти максимум . . . . .	319
5.22.2	MIN - Найти минимум . . . . .	320
5.22.3	AVG - Среднее значение . . . . .	321
5.22.4	SUM - Сумма . . . . .	323
5.22.5	APR - Математические вычисления . . . . .	325
5.22.6	PID - PID-регулирование . . . . .	328
<b>5.23</b>	<b>Логические команды . . . . .</b>	<b>336</b>
5.23.1	COM - Дополнение . . . . .	336
5.23.2	ANDW - Логическое И . . . . .	336
5.23.3	ORW - Логическое ИЛИ. . . . .	337
5.23.4	XORW - Исключающее ИЛИ . . . . .	337
5.23.5	XNRW - Исключающее ИЛИ-НЕ . . . . .	338
<b>5.24</b>	<b>Управление подпрограммами и прерываниями . . . . .</b>	<b>339</b>
5.24.1	Подпрограммы . . . . .	339
5.24.2	Прерывания . . . . .	339
5.24.3	SBS - Вызов подпрограммы . . . . .	342
5.24.4	SBN и RET - Начало подпрограммы и возврат из подпрограммы. . . . .	344
5.24.5	MCRO - Функция Макро . . . . .	344
5.24.6	INT - Управление прерываниями . . . . .	346
<b>5.25</b>	<b>Команды секции STEP . . . . .</b>	<b>350</b>
5.25.1	STEP и SNXT - Задание секции STEP / Запуск секции STEP . . . . .	350
<b>5.26</b>	<b>Специальные команды . . . . .</b>	<b>358</b>
5.26.1	FAL и FALS - Предупреждение о нефатальной ошибке и Предупреждение о фатальной ошибке . . . . .	358
5.26.2	SCAN - Время цикла . . . . .	359
5.26.3	TRSM - Выборка памяти трассировки . . . . .	359
5.26.4	MSG - Сообщение . . . . .	361
5.26.5	LMSG - Длинное сообщение . . . . .	362
5.26.6	TERM - Режим TERMINAL . . . . .	363
5.26.7	WDT - Обновление контрольного таймера . . . . .	364
5.26.8	IORR - Обновление входов/выходов . . . . .	364
5.26.9	MPRF - Обновление Модулей группы 2 . . . . .	365
5.26.10	BCNT - Счетчик битов . . . . .	366
5.26.11	FCS - Контрольная сумма кадра . . . . .	366
5.26.12	FPD - Поиск точки сбоя . . . . .	368
5.26.13	SRCH - Поиск данных . . . . .	372
5.26.14	XDMR - Чтение расширенной DM . . . . .	374
5.26.15	IEMS - Косвенная адресация EM. . . . .	375
5.26.16	EMBC - Выбор банка EM . . . . .	376
5.26.17	CMCR - Работа с картой PCMCIA . . . . .	376
<b>5.27</b>	<b>Команды связи . . . . .</b>	<b>382</b>
5.27.1	SEND - Передача по сети . . . . .	382
5.27.2	RECV - Прием из сети . . . . .	384
5.27.3	Общие сведения о сетях . . . . .	385
<b>5.28</b>	<b>Команды последовательной связи . . . . .</b>	<b>389</b>
5.28.1	RXD - Прием . . . . .	389
5.28.2	TXD - Передача . . . . .	390

5.28.3 STUP - Изменение установочных параметров RS-232C . . . . .	392
5.28.4 PMCR - Функция поддержки протоколов . . . . .	394
<b>5.29 Дополнительные команды ввода/вывода . . . . .</b>	<b>396</b>
5.29.1 DSW - Ввод с цифрового переключателя . . . . .	396
5.29.2 TKY - Ввод с клавиатуры на 10 клавиш . . . . .	399
5.29.3 HKY - Ввод с клавиатуры на 16 клавиш . . . . .	401
5.29.4 MTR - Ввод матрицы . . . . .	404
5.29.5 7SEG - Вывод на 7-сегментный индикатор . . . . .	406
<b>5.30 Команды для работы со Специальными Модулями . . . . .</b>	<b>410</b>
5.30.1 IORD - Чтение Специальных модулей . . . . .	410
5.30.2 IOWR - Запись в Специальные модули . . . . .	411
<b>Приложение А . . . . .</b>	<b>413</b>
Стандартные модели . . . . .	413
Программирующие устройства . . . . .	424
Дополнительные устройства (опции) . . . . .	424
Монтажные рейки и приспособления . . . . .	425
Программное обеспечение . . . . .	425
Материалы для обучения . . . . .	425
<b>Приложение В . . . . .</b>	<b>427</b>
Команды релейно-контактного программирования. . . . .	427
Команды C200HX/C200HG/C200HE-E . . . . .	427
Команды C200HX/C200HG/C200HE-E . . . . .	434
<b>Приложение С . . . . .</b>	<b>443</b>
Операции с флагами ошибок и арифметики . . . . .	443
<b>Приложение D . . . . .</b>	<b>449</b>
Бланки для записи распределения слов . . . . .	449
<b>Приложение E . . . . .</b>	<b>451</b>
Мнемокод программы. . . . .	451
<b>Приложение E . . . . .</b>	<b>453</b>
Таблицы преобразования данных . . . . .	453
<b>Приложение G . . . . .</b>	<b>455</b>
Расширенные коды ASCII . . . . .	455
<b>Глоссарий . . . . .</b>	<b>457</b>



## **О данной инструкции:**

Данная инструкция описывает работу программируемых контроллеров C200HX/HG/HE и включает далее перечисленные главы. Информация по установке приведена в Руководстве по установке программируемого контроллера C200HX/HG/HE. Таблица, в которой перечислены другие инструкции, которые можно использовать с данной инструкцией, приведены в Главе 1, Введение. В Главе 2, Аппаратная часть, описываются различия между ЦПУ C200HS и новыми ЦПУ, описанными в данной инструкции.

Перед тем, как начать установку и работу с C200HX/HG/HE внимательно ознакомьтесь с данной инструкцией. Обязательно ознакомьтесь с разделом “Предосторожности” в следующей главе.

**В главе 1, Введение** объясняются основы и некоторые базовые термины, используемые в языке релейно-контактного программирования. Приведены также общие сведения о процессе программирования и работе ПК и основная терминология, используемая с ПК OMRON. Кроме того, приведены описания периферийных устройств, используемых с C200HX/HG/HE и таблица других инструкций, которые можно использовать с данной инструкцией при решении специальных задач.

**В главе 2, Аппаратные средства** поясняются основные аспекты общей конфигурации ПК, описывает индикаторы, которые упоминаются в других главах данной инструкции, а также использование кассеты памяти для управления данными в UM и IOM.

**В главе 3, Области памяти** показано распределение памяти и рассказано, как пользоваться информацией, приведенной для помощи в программировании. Объясняется, как входы/выходы управляются в памяти и как биты в памяти соответствуют конкретным входам/выходам. Приведена информация о системе DM, специальной области в C200HX/HG/HE, которая позволяет осуществлять гибкое управление ПК с помощью параметров ПК.

**В главе 4, Написание и ввод программ** поясняются основы релейно-контактного программирования, описываются элементы релейно-контактной программы и то, как управлять исполнением данной программы. Объясняется также, как преобразовать релейно-контактную программу в мнемокод, чтобы ее можно было ввести с программатора.

**В главе 5, Набор команд** описаны все команды, используемые в программировании.

**В главе 6, Синхронизация при исполнении программы** описаны процессы цикла, служащие для исполнения программы и рассказано, как координировать ввод и вывод, чтобы они появлялись в должное время.

**В главе 7, Отладка и исполнение программы** описаны процедуры программатора, служащие для ввода и отладки программы, а также просмотра и управления отработкой программы.

**В главе 8, Связь** дан общий обзор характеристик систем связи C200HX/HG/HE.

**В главе 9, Поиск неисправностей** приведена информация об индикации ошибок и других средствах для сокращения времени простоя. Информация данной главы полезна также при отладке программ.

**В главе 10, Команды HOST LINK** описаны Команды HOST LINK, используемые для связи HOST LINK через порты C200HX/HG/HE.

**В Приложениях** приведены таблицы стандартных изделий OMRON, используемых с C200HX/HG/HE сводные таблицы команд, бланк записи кодов при программировании и другая информация, полезная при работе с программируемыми контроллерами.

**Предупреждение!** Если приступить к работе с ПК, не ознакомившись с данной инструкцией, то это может привести к несчастным случаям с персоналом, повреждению и поломке изделия. Пожалуйста, прочтите каждую главу перед тем, как проделать любую операцию.





## **Меры предосторожности**

*В данной главе приведены общие меры предосторожности при работе с ПК и сопутствующими устройствами.*

*Информация данной главы важна для обеспечения безопасности и надежности работы ПК. Вы обязаны прочесть и усвоить информацию данной главы перед запуском или работой с ПК.*

## 1. Для кого предназначена данная инструкция

Данная инструкция предназначена для следующего персонала, имеющего знания по электрическим системам (инженер-электрик):

- Персонал, ответственный за установку систем промышленной автоматизации.
- Персонал, ответственный за разработку систем промышленной автоматизации.
- Персонал, ответственный за обслуживание систем промышленной автоматизации.

## 2. Общие меры предосторожности

Пользователь должен обращаться с изделиями OMRON в соответствии с данной инструкцией.

Перед использованием изделий OMRON в условиях, отличающихся от описанных в инструкции или использовании их в системах управления, связанных с радиоактивными материалами, в системах управления железными дорогами, транспортом, медицинским оборудованием, аттракционами и т. д., которые могут оказать серьезное влияние на жизнь и материальные ценности в случае неправильного использования, консультируйтесь с Вашим представителем OMRON.

Номинал и характеристики изделий должны быть достаточными для систем, машин и оборудования. Обеспечьте системы, машины и оборудование механизмами двойной надежности.

В данной инструкции содержится информация для программирования и работы с ПК OMRON. Обязательно прочитайте данную инструкцию перед использованием программ и держите ее под рукой для справок при работе.

**Предупреждение!** Чрезвычайно важно, чтобы ПК и Модули ПК использовались только для тех целей и в тех условиях, для которых они предназначены, особенно когда их применение может прямо или косвенно воздействовать на жизнь. Вы должны проконсультироваться с Вашими представителями OMRON перед применением системы ПК для вышеуказанных задач.

## 3. Меры предосторожности по безопасности

**Предупреждение!** Не пытайтесь разобрать Модуль при поданном питании. Это может повлечь электрошок или смерть от поражения током.

**Предупреждение!** Не касайтесь клемм при поданном питании. Это может повлечь электрошок или смерть от поражения током.

## 4. Требования к окружающей среде

Запрещена работа системы управления в следующих условиях:

- Когда ПК подвергается прямому солнечному свету.
- Когда температура окружающей среды ниже 0° или выше 55° C.
- Когда на ПК может повлиять конденсация из-за резких изменений температуры.
- Когда окружающая влажность менее 10% или более 90%.
- При присутствии коррозионного или воспламеняющегося газа.
- При повышенной запыленности, наличии в воздухе соли или металлического порошка.
- В местах, где ПК подвергается вибрации или толчкам.
- Где на ПК может попасть вода, масло или химикаты.

**Внимание!** Окружающие условия могут сильно повлиять на срок службы и надежность системы. Неподходящие окружающие условия могут привести к сбоям, поломкам и т. д. в системе. Окружающие условия должны находиться в

допустимых диапазонах при инсталляции и должны поддерживаться в течении срока службы системы.

### 5. Меры предосторожности при применении

При применении ПК соблюдайте следующие меры предосторожности

**Предупреждение!** Несоблюдение следующих мер предосторожности может привести к серьезным или фатальным последствиям. Всегда их соблюдайте.

- Установленная система должна быть заземлена проводом с сопротивлением менее 100 Ом для защиты от поражения электротоком.
- Всегда выключайте питание перед следующими операциями. Невыполнение этого может привести к поражению электротоком.

Монтаж или демонтаж любого Модуля (т.е. Модулей входа/выхода, Модуля ЦПУ и т. д.) или Кассеты Памяти.

Сборка устройства или панели.

Подключение или отключение проводов и кабелей.

**Внимание!** Несоблюдение следующих мер предосторожностей может привести к сбоям в работе ПК или системы или может повредить ПК или Модули. Всегда их соблюдайте.

- Используйте только те блоки питания, параметры которых соответствуют указанным в Инструкции к Модулю. Другое питание может повредить Модули.
- Если питание нестабильно, примите меры, чтобы оно соответствовало номинальному.
- Для защиты от коротких замыканий на внешних подключениях ставьте размыкатели и применяйте другие меры безопасности.
- На Модулях входа не применяйте напряжение, превышающее номинальное. Это может повредить Модуль.
- С Модулями выхода не применяйте напряжение, превышающее максимально напряжение переключения. Это может повредить Модуль.
- При проведении теста на сопротивлению напряжению всегда отключайте клемму LG (зануление).
- Устанавливайте все Модули согласно инструкции. Неправильная установка может привести к некорректной работе.
- Обеспечивайте соответствующее экранирование при установке в следующих условиях:
  - В местах, которые подвергаются воздействию статического электричества или других источников помех.
    - В местах, которые подвергаются воздействию сильных электромагнитных полей.
    - В местах, которые подвергаются воздействию радиации.
    - Возле линий питания.
- Винты на Базовой Панели должны быть затянуты и кабели надежно закреплены.
- Не пытайтесь разбирать Модули, ремонтировать их или модифицировать.

**Внимание!** Следующие меры предосторожности необходимы для обеспечения общей безопасности системы. Всегда их соблюдайте.

- Обеспечивайте дублирующие механизмы для обработки некорректных сигналов, которые могут появиться из-за обрыва сигнальных линий или кратковременных прерываний питания.
- Для обеспечения безопасности создавайте внешние цепи блокировки, конечные ограничители и другие цепи безопасности в дополнение к имеющимся внутри ПК.

## **6 Соответствие директивам ЕС**

Для соответствия директивам ЕС при установке C200HX/HG/HE соблюдайте следующие меры предосторожности.

- Для источника питания постоянного тока, подключаемого к Модулю входов/выходов постоянного тока и для Модуля питания обеспечивайте усиленную изоляцию или двойную изоляцию.
- Для Модуля релейных выходов применяйте отдельный источник питания.

# 1. Введение

*В данной главе приведен краткий обзор истории программируемых контроллеров и объясняются термины, используемые в релейно-контактном программировании. Также приведен общий обзор процесса программирования и работы с ПК и базовая терминология, применяемая OMRON. Кроме того, имеются описания периферийных устройств, используемых с C200HX/HG/HE, список других инструкций, которыми следует пользоваться при специальных задачах, описание новых функций C200HX/HG/HE.*

## 1.1 Общие сведения

Программируемый контроллер (ПК) - это Центральное процессорное устройство (ЦПУ), содержащее программу, которое подключено к устройствам входа/выхода. Программа управляет ПК таким образом, что при появлении входного сигнала на входном устройстве ПК реагирует соответствующим образом. Реакция нормально заключается во включении в ON выходного сигнала на выходном устройстве. Входными устройствами могут служить фотоэлектрические датчики, кнопки на панели управления, конечные выключатели и другие устройства, которые могут выдавать сигналы, служащие входами для ПК. Выходными устройствами могут служить соленоиды, переключатели, подключающие индикаторные лампы, реле включения двигателей или любые другие устройства, которые могут управляться выходными сигналами от ПК.

Например, датчик обнаружения движущихся изделий включает в ON вход ПК. ПК в ответ включает в ON выход, который включает толкатель, который толкает изделие на другой конвейер для последующей обработки. Другой датчик, расположенный выше первого, включает в ON другой вход, указывающий, что изделие слишком высокое. ПК в ответ включает другой толкатель, расположенный перед первым толкателем, для сталкивания слишком высокого изделия в ящик для брака.

Хотя данный пример охватывает только два входа и два выхода, он типичен для показа типов операций, которые может проводить ПК. По сути, даже этот пример намного сложнее, чем может показаться на первый взгляд, из-за необходимости синхронизации, т.е. "Как ПК определяет, в какой момент задействовать каждый толкатель?". Проблема заключается в том, как получить требуемые сигналы с имеющихся входов в нужные моменты времени.

Для обеспечения управления ПК использует логику ПК, именуемую релейно-контактным программированием. Данная инструкция написана для объяснения релейно-контактного программирования и обучения читателя программированию и работе с ПК.

## 1.2 Происхождение логики ПК

Исторически ПК происходят от релейных систем управления. И хотя место дискретных реле, таймеров, счетчиков и подобных устройств заняли интегральные микросхемы и внутренняя логика ПК, операции ПК проводятся таким образом, как будто данные дискретные устройства остались на своем месте. Однако управление с помощью ПК предоставляет возможности и точность компьютера для достижения гораздо большей гибкости и надежности, чем могли бы предоставить реле.

Символика и другие понятия управления, служащие для описания операций ПК, также произошли от релейного управления и образуют основу метода релейно-контактного программирования. Однако большинство терминов, служащих для описания данных символов и понятий, происходят из компьютерной терминологии.

### Сравнение терминологии реле и ПК

Терминология, применяемая в данной инструкции, несколько отличается от терминологии реле, но идеи одни и те же.

В следующей таблице показано соотношение между терминами реле и терминами, используемыми для ПК OMRON.

Терминология реле	Эквивалент ПК
контакт	вход или условие
катушка/обмотка реле	выход или рабочий бит
нормально разомкнутый контакт	нормально открытое условие
нормально замкнутый контакт	нормально закрытое условие

На самом деле полного равенства между этими терминами нет. Термин "Условие" служит только для описания релейно-контактных программ в целом и применим только для набора базовых команд. Термины вход и выход используются в программировании только для обозначения битов входа/выхода, которые привязаны к сигналам входа и

выхода, поступающих и отходящих от ПК. Нормально открытое и нормально закрытое условие объясняются в гл. 4.4.

### 1.3 Терминология ПК

Хотя следующие термины и приведены в глоссарии в конце данной инструкции, но поскольку они являются базовыми для понимания работы ПК, они объясняются в данной главе.

#### ПК

Поскольку C200HX/HG/HE - это несколько Модулей, собранных на Панели, такого изделия как C200HX/HG/HE не существует. Вот почему мы говорим о конфигурации ПК, ПК - это собранные воедино несколько Модулей.

Для получения работающего ПК необходима Панель ЦПУ, на которой смонтирован по крайней мере один Модуль, на котором есть входы/ выходы. Когда мы имеем в виду ПК, мы как правило говорим о ЦПУ и о всех модулях, которыми оно непосредственно управляет по программе. Сюда не включаются устройства, подключенные к входам и выходам ПК.

Если Вам незнакома данная терминология, см. гл. 2.

#### Входы и выходы

Устройство, подключенное к ПК и посылающее сигнал в ПК, называется **входным устройством**. Посылаемый сигнал называется **входным сигналом**. Сигнал поступает в ПК через клеммный зажим или контакт разъема на блоке. Место подключения сигнала к ПК называется **точкой входа**. Данной точке входа присвоена ячейка памяти, которая отражает ее состояние, т.е. ON или 0. Данная ячейка памяти называется входной бит. ЦПУ в нормальном рабочем цикле просматривает состояние всех точек входа и включает в ON или 0 соответствующие входные биты.

В памяти имеются также **выходные биты**, привязанные к **выходным точкам** на блоках, через которые выходные сигналы посылаются на **выходные устройства**. Т.е. выходной бит включается в ON для посылки сигнала на выходное устройство через выходную точку. ЦПУ периодически устанавливает выходные точки в ON или 0 в соответствии с состоянием выходных битов.

Данные термины используются при описании различных аспектов работы ПК. При программировании речь идет об информации в памяти и к каким битам производятся обращения. Когда разговор идет о модулях, подключающих к ПК к управляемой системе, имеются в виду точки входов/выходов. При подключении проводов к этим точкам имеются в виду физические контакты, либо ножки контактов, либо клеммы разъемов. Когда речь идет о сигналах, которые поступают на ПК и посылаются с ПК, имеются в виду входные и выходные сигналы, или просто входы и выходы. Все это зависит от аспектов работы ПК, которые рассматриваются.

#### Управляемая система и управляющая система

Управляющая система включает ПК и все устройства входов/выходов, которые она использует для управления внешней системой. Датчик, который выдает информацию для осуществления управления, является входным устройством и составной частью системы управления. Управляемая система - это внешняя система, управляемая по программе ПК через эти устройства входов/выходов. Устройства входов/выходов можно иногда рассматривать как часть управляемой системы, например, двигатель, использующийся для привода конвейера.

### 1.4 Терминология продукции OMRON

Продукция OMRON делится на несколько функциональных групп, которые имеют "родовое" имя. В Приложении А (стандартные модели) изделия перечисляются согласно этим группам. Термин Модуль служит для обозначения всех изделий OMRON. Хотя Модуль - это составная часть, которые образуют C200HX/HG/HE, его значение обычно (но не всегда) ограничен значением составной части, которая установлена на панели.

Большинство, но не все из этих изделий, имеют имена, которые начинаются со слова Модуль.

Самая большая группа изделий OMRON - это Модули входов/выходов. Они включают все модули, установленные на панели, которые имеют неспециальные точки входов или выходов для общего пользования. модули входов/выходов поставляются с различными разъемами и характеристиками.

Модули входов/выходов высокой плотности используются для обеспечения возможности высокой плотности входов/выходов и включают модули входов/выходов высокой плотности, группа-2 и специальные модули входов/выходов высокой плотности.

Специальные Модули используются для решения особых задач. Они включают некоторые модули входов/выходов высокой плотности, модули позиционирования, модули высокочастотного счетчика и модули аналоговых входов/выходов.

Коммуникационные Модули используются для создания систем связи, которые объединяют несколько ПК или соединяют ПК с удаленными точками входов/выходов. модули связи включают модули удаленных входов/выходов, модули связи PC LINK, модули связи с управляющим устройством HOST LINK, модули связи SYSMAC NET.

Другие группы изделий включают программирующие устройства, периферийные устройства и профиля DIN RAIL.

### 1.5 Краткое описание работы ПК

Далее описаны основные операции, которые используются в программировании и работе C200HX/HG/HE. Предполагая, что Вы уже купили один или несколько ПК, Вы, вероятно, имеете достаточное представление о шагах 1 и 2, и они здесь описаны лишь вкратце. Данная инструкция написана, чтобы объяснить шаги 3.6, 8, 9 из перечисленных далее. Соответствующие главы данной инструкции предоставляют подробную информацию о данных операциях.

- 1, 2, 3,...** 1. Определите, что должна делать управляемая система, в каком порядке и в какое время.
2. Определите, какие панели и модули потребуются. См. Руководство по установке C200HX/HG/HE. Если требуется система связи. см. инструкцию по работе с соответствующей системой.
3. На бумаге распределите все устройства входов/выходов по точкам на блоках входов/выходов и определите, какие биты будут присвоены каждому устройству. Если в ПК включены модули специальных входов/выходов или системы связи, подробности о распределении битов см. инструкцию по этим устройствам.
4. Используя релейно-контактные символы, напишите программу, которая представляет собой последовательность требуемых операций и их взаимосвязи. Обеспечьте также соответствующие программы для аварийных ситуаций. (см. главы 4, 5 и 6).
5. Введите программу и все требуемые рабочие параметры в ПК (см. 4.7).
6. Отладьте программу, сперва уберите синтаксические ошибки, и затем найдите ошибки при работе. ( см. 4.7, 7 и 9).
7. Подключите провода от управляемой системы к ПК. Данный шаг можно производить после завершения шага 3. Подробности о подключении каждого блока см. Инструкцию по установке C200HX/HG/HE.
8. Прогоните программу в реальной ситуации и выполните точную настройку. (см. 7 и 9).
9. Сделайте две копии отлаженной программы и храните их в разных местах. (см. 4.7).

#### Разработка системы управления

Разработка системы управления - это первый шаг в автоматизации любого процесса. ПК можно программировать только после полного понимания системы управления.



Разработка системы управления требует прежде всего полного понимания того оборудования, над которым осуществляется управление. Таким образом, первый шаг разработки - определение требований управляемой системы.

### **Требования к входам/выходам**

Прежде всего следует определить количество точек входов/выходов, требуемое для системы. Это достигается идентификацией каждого устройства, который посылает сигнал на ПК или получает выходной сигнал от ПК. Помните, что количество имеющихся точек входов/выходов зависит от конфигурации ПК. Подробности о количестве входов/выходов и распределении битов и входов/выходов см. 3.3.

### **Последовательность операций, синхронизация и взаимосвязи**

Далее определяется последовательность операций управления во времени. Указываются физические соотношения между устройствами входов/выходов и их реакция.

Например, фотоэлектрический датчик может быть функционально привязан к двигателю с помощью счетчика внутри ПК. Когда ПК получает входной сигнал от пускового выключателя, он запускает двигатель. ПК может остановить двигатель, когда счетчик отсчитает заданное количество сигналов с фотоэлектрического датчика.

Каждая из связанных задач должна быть определена подобным образом, с начала до конца операций управления.

### **Требования к блокам**

Каждый Модуль, установленный или подключенный на панели ПК, должен быть назначен в соответствии с требованиями к устройствам входов/выходов. Следует учитывать характеристики аппаратуры, такие, как напряжение или ток, и функциональные особенности специальных модулей входов/выходов или систем связи. Во многих случаях модули специальных входов/выходов, модули интеллектуальных входов/выходов или системы связи могут значительно сократить программирование. Более подробно эти модули и системы связи описаны в соответствующих инструкциях по работе.

После разработки системы управления в целом можно начинать выполнение следующих задач: программирование, отладка и другие операции, описанные в данной инструкции.

## 1.6 Периферийные устройства

При программировании можно использовать следующие периферийные устройства для ввода/отладки/просмотра программы ПК или подключения ПК к внешним устройствам для вывода программы либо данных из области памяти. Номера моделей всех устройств, перечисленных далее, описаны в Приложении А. В последующих описаниях имена изделий OMRON выделены жирным шрифтом.

### **Консоль программирования**

Консоль программирования - простейшее устройство программирования для контроллеров OMRON. Все программаторы подключаются непосредственно к ПК без отдельного интерфейса.

### **Программа поддержки SYSMAC - SYSMAC-CPT (или SYSWIN)**

ПО разработана для работы на IBM PC/AT или совместимом компьютере и позволяет осуществлять все операции программатора, а также много дополнительных. Программы ПК можно написать на дисплее как в релейно-контактной форме, так и мнемокоде. Когда программа написана, она высвечивается на дисплее, позволяя легко производить изменения. Можно провести контроль синтаксиса перед загрузкой программы в ПК.

ПО поставляется на дискетах 3.5".

## 1.7 Имеющиеся инструкции

В следующей таблице перечисляются другие инструкции, которые могут потребоваться для программирования и работы с ПК C200HX/HG/HE. Инструкции по работе есть для

## 1.7 Имеющиеся инструкции

каждого отдельного блока и к ним следует обращаться при подключении и других работах.

Наименование	Номер по каталогу	Содержание
Инструкция по работе с модулем аналоговых входов/выходов	W127	Информация о модулях аналоговых входов/выходов C200H-AD001, C200H-DA001
	W229	Информация о модуле аналоговых входов C200H-AD002/DA002
	W325	Информация о модуле аналоговых входов C200H-AD003/DA003/DA004/MAD01
Инструкция по работе с модулем датчика температуры	W124	Информация о модуле датчика температуры
Инструкция по работе с модулем терморегулятора	W225	Информация о модуле терморегулятора
Инструкция по работе с модулем терморегулятора (нагрев/охлаждение)	W240	Информация о модуле терморегулятора с дискретным выходом
Инструкция по работе с модулем PID-регулятора	W241	Информация о модуле PID-регулятора
Инструкция по работе с модулем высокоскоростного счетчика	W141	Информация о модулях высокоскоростного счетчика CT001, CT002
	W311	Информация о модуле высокоскоростного счетчика CT021
Инструкция по работе с модулем позиционирования	W137	Информация о модуле позиционирования NC111
	W128	Информация о модуле позиционирования NC112
	W166	Информация о модуле позиционирования NC211
	W334	Информация о модулях позиционирования NC113, NC213, NC413
Инструкция по работе с модулем позиционирования по уставкам	W224	Информация о модуле позиционирования по уставкам (командоаппарат)
Инструкция по работе с модулем управления движением	W314, W315	Информация о модуле управления движением
Инструкция по работе с модулем ASCII	W165	Информация о модуле ASCII
Инструкция по работе с модулем датчика ID	W153	Информация о модуле датчика идентификации
Инструкция по работе с модулем голоса	W172	Информация о модуле записи и воспроизведения звуковой информации
Инструкция по работе с модулем FUZZY	W208	Информация о модуле нечеткой логики (Fuzzy-логика)
Инструкция по работе с программного обеспечения для FUZZY модуля	W210	Информация о работе с ПО
Инструкция по работе с пультом доступа к данным	W173	Индикация области данных и процедуры изменения данных при работе с пультом доступа к данным
Инструкция по работе с Модулями входов/выходов высокой плотности	W133	Информация о Модулях входов/выходов высокой плотности и о смешанных Модулях
Инструкция по работе с коммуникационными панелями	W304	Информация о Панелях C200HW-COM_
Инструкция по работе с Модулем PC Card	W313	Информация о работе с модулем PC Card

## 1.8 Характеристики C200HX/HG/HE

Наименование	Номер по каталогу	Содержание
Инструкция по работе с системой связи Controller Link	W309	Информация по построению системы связи Controller Link
Инструкция по работе с системой связи CompoBus/S		Информация по построению системы связи CompoBus/S
Инструкция по работе с системой связи CompoBus/D		Информация по построению системы связи CompoBus/D
Инструкция по работе с проводной системой удаленного ввода/вывода (SYSMAC BUS)	W120	Информация по построению проводных систем удаленного ввода/вывода
Инструкция по работе с оптической системой удаленного ввода/вывода (SYSMAC BUS)	W136	Информация по построению оптической системы удаленного ввода/вывода
Инструкция по работе с системой связи PC LINK	W135	Информация по построению системы связи PC LINK для автоматического обмена данными между ПК
Инструкция по работе с системой связи HOST LINK (SYSMAC WAY)	W143	Информация по построению системы связи HOST LINK для управления ПК с ведущего компьютера
Инструкция по работе с модулем связи SYSMAC NET	W114	Информация по построению системы связи SYSMAC NET для создания оптической локальной сети (LAN), включающей ПК с компьютерами и другими периферийными устройствами
Инструкция по работе с системой связи SYSMAC LINK	W174	Информация по построению системы связи SYSMAC LINK, позволяющей автоматическую передачу данных, программирование и программируемый обмен данными между ПК в системе

## 1.8 Характеристики C200HX/HG/HE

C200HX/HG/HE имеют несколько новых особенностей, однако программы C200H и C200HS можно использовать на новых ЦПУ.

В Модулях ЦПУ C200HX/HG/HE - ZE имеется набор команд, который шире, чем в базовых Модулях ЦПУ C200HX/HG/HE, и команды задаются 3-разрядными функциональными кодами.

### 1.8.1 Усовершенствования в C200HX/HG/HE - ZE

#### Команды

В таблице приведены дополнительные команды Модулей ЦПУ C200HX/HG/HE -ZE.

Команда	
Команды управления битами	BIT TEST (Тест бита): TST(350) и TSTN(351)
Специальные команды	EM BANK TRANSFER (Передача банка EM): BXFR(125)
Команды входного сравнения	Равенство: =(300), =L(301), =S(302), =SL(303)
	Неравенство: <>(305), <>L(306), <>S(307), <>SL(308)
	Меньше чем: <(310), <(311), <S(312), <SL(313)
	Меньше чем или равно: <=(315), <=L(316), <=S(317), <=SL(318)
	Больше чем: >(320), >L(321), >S(322), >SL(323)
	Больше чем или равно: >=(325), >=L(326), >=S(327), >=SL(328)
Символьные команды математики	Двоичное сложение: +(400)/ +L(401)/+C(402)/+CL(403)
	Двоично-десятичное сложение: +B(404)/ +BL(405)/+BC(406)/+BCL(407)
	Двоичное вычитание: -(410)/ -L(411)/-C(412)/-CL(413)

Команда	
	Двоично-десятичное вычитание: -B(414)/ -BL(415)/-BC(416)/-BCL(417)
	Двоичное умножение: *(420)/*L(421)/*U(422)/*UIL(423)
	Двоично-десятичное умножение: *B(424)/*BL(425)
	Двоичное деление: /(430)/ /L(431)/ /U(432)/ /UL(433)
	Двоично-десятичное деление: /B(434)/ /BL(435)

### Другие усовершенствования

В Модулях ЦПУ C200HX/HG/HE - ZE сделаны также следующие усовершенствования:

- Коды функций стали трехзначными, поэтому почти все команды можно использовать без присвоения функциональных кодов.
- Можно использовать пакет поддержки SYSMAC-CPT, который намного расширяет возможности программирования/просмотра. (Предыдущие пакеты поддержки нельзя использовать)
- К адресам области EM может быть прямая адресация в операндах команд. Это практически удваивает память слов, к которой может быть прямая адресация из программы (6 655 слов области DM плюс 6 143 слова области EM).
- Максимальный объем памяти пользователя удвоился с 32 К слов до 64 К слов.

### 1.8.2 Возможности C200HS и C200HX/HG/HE

В таблице приведены новые возможности C200HX/HG/HE и их сравнение с C200HS.

Функция		Характеристика	
		C200HX/HG/HE	C200HS
Память	Память пользователя (UM) К слов	C200HX-CPU11-E: 3.2 C200HE-CPU_2-E: 7.2 C200HG-CPU_3-E: 15.2 C200HX-CPU_4-E: 31.2	15.2
	Нормальная DM, слов	6 144 (DM 0000..DM 6143) (У C200HX-CPU11-E нет DM 4000..DM 5999)	6 144 (DM 0000..DM 6143)
	Фиксированная DM, слов	512 (DM 6144..DM 6655)	512 (DM 6144..DM 6655)
	Дополнительная DM, слов	0..3 000 (DM 7000..DM 9999)	0..3 000 (DM 7000..DM 9999)
	Дополнительная память данных (EM), слов	6 144 (EM 0000..EM 6143) C200HE: нет C200HG: 6 1444 слов x 1 банк C200HX: 6 1444 слов x 3 банка	Нет
Распределение входов/выходов	Панели расширения	3 панели (2 панели в C200HE-CPU__-E или C200HX/HG-CPU3_-E/4_-E)	2 панели
	Модули входов/ выходов группы 2	Номера модулей 0..9, A..F (Несовместимы с C200HE-CPU11-E) (Номера модулей 0..9 в C200HE-CPU__-E или C200HX/HG-CPU3_-E/4_-E).	Номера модулей 0..9
	Модули специальных входов/выходов	Номера модулей 0..9, A..F (Номера модулей 0..9 в C200HE-CPU__-E или C200HX/HG-CPU3_-E/4_-E)	Номера модулей 0..9
Время исполнения	Базовые команды (LD), мкс	0.104 (C200HX) 0.156 (C200HG) 0.313 (C200HE)	0.375

## 1.8 Характеристики C200HX/HG/HE

Функция		Характеристика	
		C200HX/HG/HE	C200HS
	MOV, мкс	0.417 (C200HX) 0.625 (C200HG) 1.250 (C200HE)	19.00 мкс
	ADD, мкс	16.65 (C200HX/HG) 31.45 (C200HE)	40.10 мкс
	Другие команды	C200HX/HG: 1/3..2/3 времени C200HS C200HE: 3/4..4/5 времени C200HS	-
	Общие процессы (обработка END), мс	0.7 (C200HX/HG) 2.1 (C200HE)	0.7
	Время обновления входов/выходов	То же, что и в C200HS, хотя часть специальных входов/выходов требует 1/2..1/3 времени C200HS	-
Функции ЦПУ	Порт RS-232C	Есть в C200HX/HG/HE-CPU4_-E/6_-E	Есть в C200HS-CPU2_-E/3_-E
	Функция часов	Есть у всех, кроме C200HE-CPU11-E	Есть во всех моделях
	SYSMAC NET и SYSMAC LINK	Панели связи можно установить на все ПК, за исключением C200HE-CPU11-E. (Номера моделей плат C200HW-COM01/04-E)	Есть в 200HS-CPU3_-E
Панели связи		Панели связи можно установить на все ПК, за исключением C200HE-CPU11-E	-
Модули специальных входов/выходов		Команды IORD и IOWR позволяют передавать данные на и с модулей специальных входов/выходов	-
Прерывания	Модули входов прерываний	2 модуля (16 входов)	1 Модуль (8 входов)
	Прерывания панели связи	Можно задать	-
	Время реакции	То же, что и в C200HS, хотя возможна реакция 1 мс в C200HW-SLK__	Нормальный режим: 10 мс Высокоскоростной режим: 1 мс (Всегда 10 мс при использовании SYSMAC NET и SYSMAC LINK)
Программируемые терминалы (ПТ)		NT Link (1:1) или (1:N) Можно подключить до 8 ПТ с порта RS-232C через адаптер связи RS-422/485. При использовании панели связи с C200HE-CPU__-E можно подключить только 3 ПТ	NT Link (1:1)
SYSMAC LINK	Время обслуживания	3.5 мс макс. (1 рабочий уровень)	10.8 мс макс, (1 рабочий уровень)
	Удаленное программирование	Возможно с периферийного порта или порта RS-232C (включая панель связи)	Возможно с периферийного порта
	Влияет на время ответа	Нет	10 мс в любом режиме

### 1.8.3 Совместимость программ

Программы C200HS и кассеты памяти можно применять так же, как и в C200HX/HG/HE. программы, разработанные для C200H можно передавать для использования в C200HX/HG/HE.

Программы, разработанные для C200H, C200HS и C200HX/HG/HE можно использовать и в C200HX/HG/HE-ZE.

Подробное описание отдельных процедур для передачи программ есть в инструкции по работе с ПО.

#### Предосторожности при работе с C200HX/HG/HE-CPU\_\_-E

При передаче программ на C200HX/HG/HE соблюдайте следующие предосторожности:

- Если программа C200H включает команду SYS и передается в C200HX/HG/HE, рабочие параметры, заданные этой командой, будут переданы в область установочных параметров C200HX/HG/HE (DM 6600, DM 6601 и 6655) и перепишет текущие значения. Перед использованием программы после передачи обязательно убедитесь, что параметры заданы корректно.
- Если программа в C200H обращается к протоколу ошибок по адресам DM 0969..DM 0999, адреса следует изменить на DM 6000..DM 6030, что является областью протокола для C200HX/HG/HE.
- Программы, которые зависят от времени цикла (т.е. от времени, требуемого для выполнения части программы) должен быть настроен на время C200HX/HG/HE, цикл в котором гораздо короче.

#### Предосторожности при работе с C200HX/HG/HE-CPU\_\_-ZE

Соблюдайте следующие предосторожности при использовании Модуля ЦПУ C200HX/HG/HE - ZE.

- Строки комментария в программах для C200HX/HG/HE - ZE, которые созданы с помощью пакета поддержки SYSMAC-CPT, сохраняются в командах CMT (команды, которые сохраняют информацию о позиции строки комментария). Таким образом, размер программ C200HX/HG/HE со строками комментария увеличивается, если программы копируются в Модуль ЦПУ C200HX/HG/HE - ZE с помощью пакета поддержки SYSMAC-CPT. (Команды CPT будут индигироваться на консоли программирования, но их нельзя добавить, изменить или удалить с консоли программирования).
- Модули ЦПУ C200HX/HG/HE - ZE не поддерживают форму фронта 0/1 команды TEN KEY (@TKY), хотя их можно ввести с SYSMAC-CPT. Не используйте эту команду.
- Будьте внимательны при преобразовании команд расширения из программ C200HX/HG/HE при импорте, копировании или добавлении в программы C200HX/HG/HE-ZE. Если используются функциональные коды команд расширения по умолчанию, они автоматически будут преобразованы в 3-разрядные коды. Команды, перечисленные в правом столбце следующей таблицы, не будут преобразованы корректно и появится сообщение об ошибке (отсутствует функциональный код для команд). Либо используйте команды из правого столбца, или скорректируйте программу после преобразования. (Даже команды, которые не преобразуются, можно использовать путем присвоения им функциональных кодов как командам расширения).

Команды, которые не преобразуются автоматически	Рекомендуемые команды
ADBL	+CL
DBS	/
DBSL	/L
MBS	*
MBSL	*L
SBBL	-CL
BXF2	BXFR (Определяет расширенное DM как операнд)

XFR2	XFER (Определяет расширенное DM как операнд)
IEMS	Нет (Возможно определение расширенного DM как операнд)

- SYSMAC-CPT нельзя подключать через Модуль Host Link.
- Если время цикла более 100 мс при он-лайнном редактировании из SYSMAC-CPT, измените время работы со связью (т.е. задержку) на 10 с.
- Для записи на Кассету памяти, установленную в ЦПУ, требуется время. При использовании SYSMAC-CPT для записи данных на Кассету памяти из памяти пользователя или памяти входов/выходов измените время работы со связью (т.е. задержку) на 15 с перед включением бита управления области SR.

#### Использование внутренней памяти

Далее описаны процедуры передачи программ C200H в память пользователя на C200HX/HG/HE.

- 1, 2, 3,...** 1. Передайте программу и требуемые данные в рабочую область ПО. Эти данные можно передавать с ЦПУ C200H, дискеты или из модуля памяти C200HS.

Для передачи с ЦПУ C200H настройте ПК для ПО на C200H, войдите в режим ON-LINE, передайте программу и требуемые данные в рабочую область ПО. Вы, вероятно, захотите передать параметры DM и таблицу входов/выходов C200H (если Вы ее создали).

или

Для передачи с дискеты, настройте ПК для ПО на C200H в режиме OF-LINE и загрузите программу и другие требуемые данные в рабочую область ПО. Вы, вероятно, захотите передать параметры DM и таблицу входов/выходов C200H (если Вы ее создали).

или

Для передачи с C200H-MP831, настройте ПК для ПО на C200H в режиме OFF-LINE и загрузите программу и другие требуемые данные с модуля памяти в рабочую область ПО.

2. Если ПО еще не в режиме OFF-LINE, войдите в режим OFF-LINE.
3. Измените установочные параметры ПО для C200HX/HG/HE.
4. Если Вы хотите передать комментарии входов/выходов совместно с программой в C200HX/HG/HE, выделите область UM для комментариев входов/выходов.
5. Подключите ПО к C200HX/HG/HE и войдите в режим ON-LINE.
6. Обеспечьте, чтобы секция 1 ЦПУ C200HX/HG/HE была в положении OFF, разрешая запись в область UM.
7. Передайте программу и другие требуемые данные в C200HX/HG/HE. Вы, вероятно, захотите передать параметры DM и таблицу входов/выходов C200H (если Вы ее создали).
8. Выключите C200HX/HG/HE и снова включите.
9. Перед рабочим режимом проведите тестовый прогон программы.

#### Использование кассет памяти

Далее описаны процедуры передачи программ C200H в C200HX/HG/HE с помощью кассет памяти EEPROM или EPROM. Это позволяет автоматически считывать программу с кассеты памяти при пуске C200HX/HG/HE. Первые 4 шага данного алгоритма те же самые, что и для передачи непосредственно в память программ C200HX/HG/HE (область UM).

- 1, 2, 3,...** 1. Передайте программу и требуемые данные в рабочую область пакета поддержки. Эти данные можно передавать с ЦПУ C200H, с дискеты или из Модуля Памяти.

Для передачи с Модуля ЦПУ C200H подключите пакет поддержки на C200H, войдите в режим ON-LINE, передайте программу и требуемые данные в рабочую область пакета поддержки. Вы, вероятно, захотите передать параметры DM и таблицу входов/выходов C200H, если Вы создали таблицу входов/выходов для C200H.

или

Для передачи с жесткого диска установите Пакет Поддержки в режим офф-лайн (автоном) и загрузите программу, параметры DM и таблицу входов/выходов, в рабочую область пакета поддержки. Их требуется загружать раздельно. Вы, вероятно, захотите загрузить параметры DM и таблицу входов/выходов, если Вы создали таблицу входов/выходов для C200H.

2. Если пакет поддержки еще не в режиме OFF-LINE, войдите в режим OFF-LINE
3. Настройте установочные параметры для пакета поддержки.
4. Настройте установочные параметры для пакета поддержки C200HX/HG/HE.
5. Если Вы хотите передать комментарии входов/выходов совместно с программой в C200HX/HG/HE, выделите область UM для комментариев входов/выходов.
6. Выделите слова расширенной памяти DM 7000..DM 7999 в области UM с помощью операции выделения UM для пакета поддержки.
7. Копируйте DM 1000..DM 1999 в DM 7000..DM 7999.
8. Запишите "0100" в DM 6602 для автоматической передачи содержимого DM 7000..DM 7999 в DM 1000..DM 1999 при пуске.
9. Для передачи в кассету памяти EEPROM проделайте следующие операции:
  - a. Подключите пакет поддержки к C200HX/HG/HE и войдите в режим ON-LINE.
  - b. Обеспечьте, чтобы секция 1 ЦПУ C200HX/HG/HE была в положении OFF, разрешая запись в область UM.
  - c. Передайте программу и другие требуемые данные в C200HX/HG/HE. Вы, вероятно, захотите передать параметры DM и таблицу входов/выходов C200H, если Вы создали таблицу входов/выходов для C200H. Убедитесь, что вы задали передачу области Расширенных DM и, при необходимости, области комментариев Входов/выходов.
  - d. Передайте программу и другие требуемые данные в C200HX/HG/HE. Вы, вероятно, захотите передать параметры DM и таблицу входов/выходов C200H, если Вы создали таблицу входов/выходов для C200H. Убедитесь, что вы задали передачу области Расширенных DM и, при необходимости, области комментариев Входов/выходов.
10. Включите в ON секцию 2 переключателя DIP ЦПУ C200HX/HG/HE для автоматической передачи данных из кассеты памяти в Модуль ЦПУ при пуске
11. Выключите C200HX/HG/HE и снова включите для сброса и передачи данных с кассеты памяти в Модуль ЦПУ.
12. Перед рабочим режимом проведите тестовый прогон программы

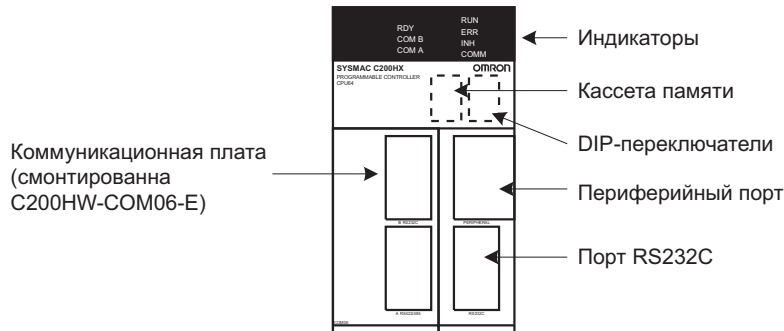


## **2. Аппаратная часть**

*В данной главе приведена информация об аппаратной части С200НХ/НГ/НЕ, которая связана с программированием и операциями программного обеспечения. Сюда включаются компоненты ЦПУ, конфигурация базового ПК, возможности ЦПУ и кассеты памяти. Данная информация изложена подробно в Инструкции по установке С200НХ/НГ/НЕ.*

## 2.1 Компоненты ЦПУ

На следующем рисунке показаны основные составные части ЦПУ.



### Кассета памяти

В ЦПУ есть отдел для подключения кассеты памяти. Кассета работает как RAM (ОЗУ) совместно с встроенной RAM ЦПУ.

### Периферийный порт

К периферийному порту можно подключить периферийное устройство.

### Порт RS-232C

В ЦПУ имеется встроенный порт RS-232C.

### Панель связи

В ЦПУ имеется отдел для подключения панели связи.

### Переключатель DIP

ПК работает в соответствии с положением секций переключателя DIP на ЦПУ. DIP для C200HX/HG/HE имеет 6 секций. Функции каждой секции описаны в таблице. (При поставке все 6 секций находятся в положении OFF).



OFF - ON

Секция	Положение	Функция
1	ON	Данные нельзя записывать в область UM
	OFF	Данные можно записывать в область UM
2	ON	Данные из Кассеты памяти автоматически читаются при пуске
	OFF	Данные из Кассеты памяти не читаются автоматически при пуске
2	ON	Сообщения на программаторе пишутся по-английски
	OFF	Сообщения на программаторе пишутся по-японски
4	ON	Дополнительные команды можно устанавливать
	OFF	Дополнительные команды нельзя устанавливать (установлены по умолчанию)
5	ON	Устанавливает следующие условия для порта связи (включая ситуацию, когда CQM1-CIF02 подключен к периферийному порту): 1 стартовый бит, 7 битов данных, контроль на четность, 2 стоповых бита, скорость обмена 9600 бод
	OFF	Отменяет вышеприведенные установки

Секция	Положение	Функция
6	ON	Консоль программирования в режиме расширенного терминала (AR 0712 = ON)
	OFF	Консоль программирования в нормальном режиме (AR 0712 = OFF)

### 2.1.1 Индикаторы ЦПУ

Индикаторы ЦПУ обеспечивают визуальную информацию о работе ПК. Хотя индикаторы не являются заменой программных средств, использующих флаги и другие указатели в памяти, индикаторы свидетельствуют о нормальной работе ЦПУ. Индикаторы описаны в таблице.

Индикатор	Функция
RUN (зеленый)	Горит при нормальной работе ЦПУ
ERR (красный)	Мигает при обнаружении нефатальной ошибки. ПК продолжает работать. Горит при обнаружении фатальной ошибки. ПК прекращает работать. После прекращения работы выключается индикатор RUN и все выходные модулей будут установлены в OFF.
INH (оранжевый)	Горит, когда флаг LOAD OFF (сбросить нагрузку, бит AR) =ON. Все выходные сигналы Модулей будут установлены в OFF
COMM (оранжевый)	Мигает, когда ЦПУ связывается с другим устройством по периферийному порту или RS-232C

### 2.1.1 Подключение периферийного устройства

Для программирования или просмотра состояния C200HX/HG/HE можно использовать Консоль программирования или IBM PC/AT с ПО (или Syswin v3.1 или выше).

#### Консоль программирования

Консоль программирования C200H-PR027-E или CQM1-PRO01-E можно подключить в соответствии со схемой. C200H-PR027-E подключается через кабель C200H-CN222 или C200H-CN422, который нужно покупать отдельно. С CQM1-PRO01-E поставляется кабель.

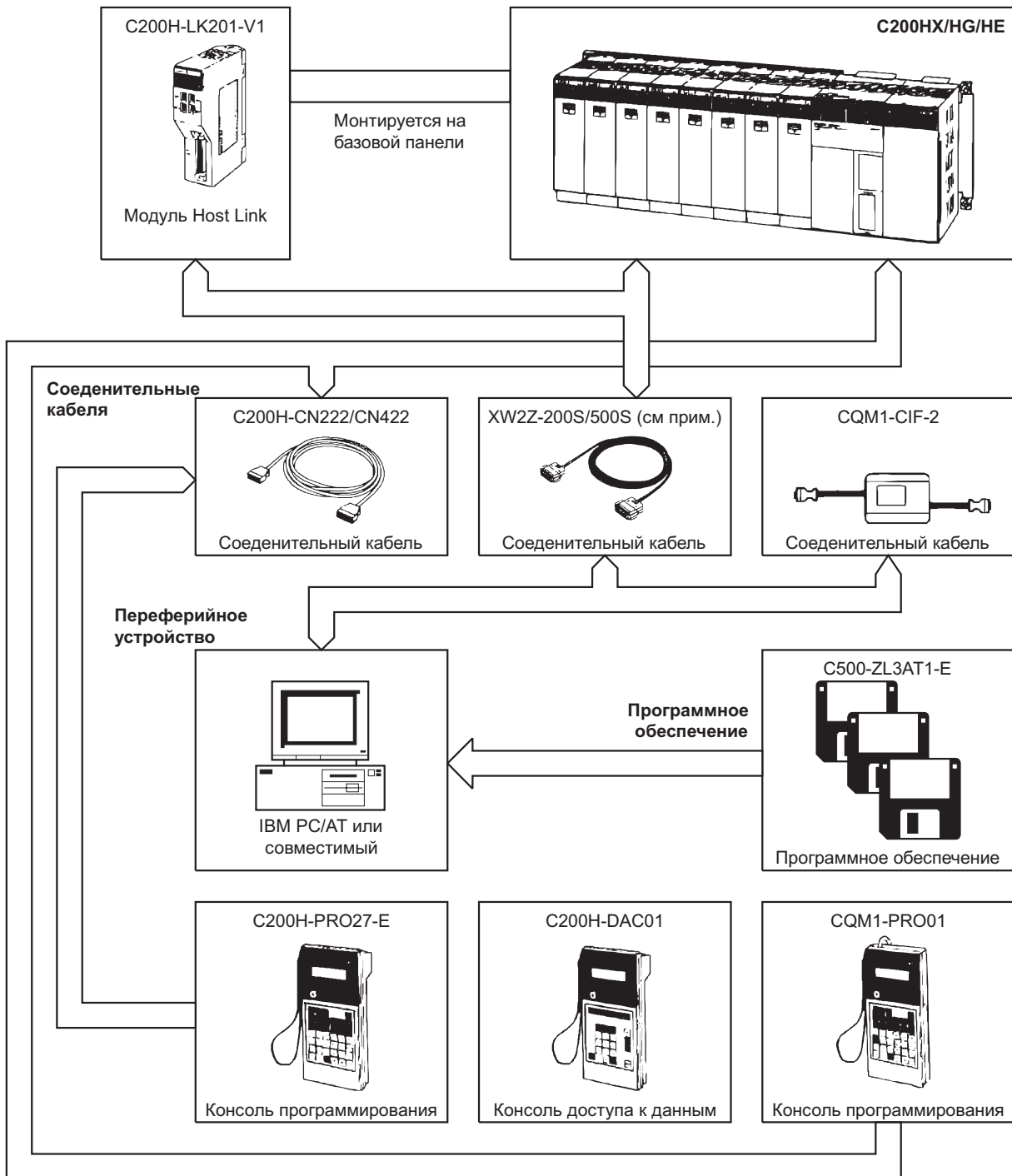
#### Консоль доступа к данным

Консоль доступа к данным C200H-DAC01 можно подключить через кабель C200H-CN222 или C200H-CN422, который нужно покупать отдельно. Когда C200H-DAC01 работает с C200HX/HG/HE, следующие операции недоступны:

- Чтение и изменение задания
- Индикация сообщений об ошибках.

**IBM PC/AT с ПО**

IBM PC/AT или совместимый компьютер с ПО подключаются в соответствии со схемой.



**Замечание** Разъем соединительного кабеля XW2Z-200S/500S - это вилка с 25 штырьками. Разъему со стороны IBM PC/AT или совместимого компьютера (9 штырьков D-SUB) требуется адаптер-переходник.

**2.2 Конфигурация ПК**

Базовая конфигурация ПК состоит из 2 типов панелей: Панель ЦПУ и Панель расширения. Панели расширения - необязательная часть базовой системы. Они используются для увеличения числа входов/выходов. Информация о таких панелях

приведена в п.3.3. Третий тип панелей, называемый Ведомыми панелями, используется, когда ПК работает с Системой Sysmac Bus.

### **Панель ЦПУ**

Панель ЦПУ С200НХ/НГ/НЕ состоит из следующих составных частей:

- Базовая панель ЦПУ, на которой монтируются ЦПУ и другие модули.
- Модуль питания
- ЦПУ, которое выполняет программу и управляет контроллером.
- Другие Модули, такие как Модули входов/выходов, Специальные Модули и Коммуникационные Модули, которые обеспечивают физические входы/выходы, соответствующие точкам входов/выходов.

Панель ЦПУ С200НХ/НГ/НЕ может использоваться отдельно или в соединении с другими панелями для увеличения числа входов/выходов. Базовые панели ЦПУ выпускаются на 3, 5, 8 или 10 платомест.

### **Панели расширения**

Панель расширения можно считать расширением ПК, поскольку она предоставляет дополнительные платоместа, на которые монтируются Модули. Панель собирается на Базовой панели расширения, на которой монтируются Модуль питания и до 10 других Модулей.

Панель расширения всегда соединяется с Панелью ЦПУ через разъемы на задней панели, которые позволяют осуществить связь между двумя панелями. К Панели ЦПУ можно последовательно подключить до трех Панелей расширения входов/выходов (две у С200НЕ).

### **Позиция установки модуля**

На Ведомой панели можно установить только Модули входов/выходов и Специальные Модули. Все Модули входов/выходов, Специальных Модули, Модули группы-2, Мастер-модули Sysmac Bus и Модули Host Link можно установить в любое платоместо на всех других панелях.

Подробности о конфигурации ПК и платоместах см. Руководство по установке С200НХ/НГ/НЕ. Распределение входов/выходов описаны в п.3.3 Область IR.

## 2.3 Характеристики ЦПУ

В следующей таблице приведены характеристики ЦПУ контроллеров C200HX/HG/HE.

Встроенный RS232C										
Разъем панели связи панели связи										Встроенный RS232C
Макс. число Специальных Модулей								Функция часов		
Макс. число Модулей группы 2						Нет	10	Нет	Нет	
Макс. число Панелей расширения				0.3	2					
Время исполнения базовых команд, мкс			Нет							
Объем EM, К слов		6								
Объем DM, К слов										
Объем памяти программ, К слов										
Объем памяти программ, К слов										
Объем памяти программ, К слов										
Объем памяти программ, К слов										
Объем памяти программ, К слов										
Объем памяти программ, К слов										
Модель	3.2	4	Нет	0.3	2	Нет	10	Нет	Нет	Нет
C200HE-CPU11-E	7.2	6	Нет	0.3	2	Нет	10	Нет	Нет	Нет
C200HE-CPU11-ZE										
C200HE-CPU32-E										
C200HE-CPU32-ZE										
C200HE-CPU42-E										
C200HE-CPU42-ZE										
C200HE-CPU42-ZE	15.2	6	6	0.15	2	10	10	Да	Да	Нет
C200HG-CPU33-E										
C200HG-CPU33-ZE										
C200HG-CPU43-E										
C200HG-CPU43-ZE										
C200HG-CPU53-E										
C200HG-CPU53-ZE										
C200HG-CPU63-E										
C200HG-CPU63-ZE										
C200HG-CPU63-ZE										
C200HX-CPU34-E	31.2	6	18	0.1	2	10	10	Да	Да	Нет
C200HX-CPU34-ZE										
C200HX-CPU44-E										
C200HX-CPU44-ZE										
C200HX-CPU54-E										
C200HX-CPU54-ZE										
C200HX-CPU64-E										
C200HX-CPU64-ZE										
C200HX-CPU64-ZE										
C200HX-CPU64-ZE										
C200HX-CPU65-ZE	63.2	6	48	0.1	2	10	10	Да	Да	Нет
C200HX-CPU85-ZE			96							

## 2.4 Кассеты памяти

C200HX/HG/HE прибывают с встроенной RAM (ОЗУ) для программы пользователя, так что нормальную программу можно создать без установки Кассеты памяти.

Дополнительная Кассета памяти нужна для сохранения программы, установочных параметров ПК, комментариев входов/выходов, области DM и другого содержимого областей данных. Подробности об установке Кассет памяти см. Руководство по установке C200HX/HG/HE.

### Функции Кассеты памяти

Кассета памяти служит для записи и чтения областей UM и IOM. UM, сохраненную на кассете, можно сравнивать с UM в RAM.

- 1, 2, 3,...** 1. Содержимое UM (память пользователя) можно сохранить в кассете памяти для последующего считывания или сравнения. Если секция 2 переключателя DIP ЦПУ = ON, содержимое кассеты памяти автоматически запишется в память при пуске ПК.

В области UM находится релейно-контактная программа, фиксированная DM (установочные параметры), расширенная DM, комментарии входов/выходов, таблица входов/выходов и информация о распределении UM.

2. Содержимое памяти входов/выходов (IOM) можно сохранить в Кассете памяти для последующего считывания.

IOM включает области IR, SR, LR, HR, AR, текущие значения счетчиков и таймеров, DM 0000..DM 6143 и EM 0000..EM 6143.

Данные в UM и IOM полностью взаимозаменяемы для C200HX/HG/HE и для C200HS, за исключением той части параметров C200HX/HG/HE, которая превышает емкость памяти C200HS и новых команд (BXF, IEMS, IORD, IOWR, PMCR, STUP, XFR2, которые не поддерживаются C200HS. Адреса области данных и команды, которые не поддерживаются C200HS, нельзя использовать на C200HS. Данные IOM нельзя записывать в RAM контроллера, если размер IOM на кассете памяти не соответствует размеру IOM на ПК.

### Совместимые кассеты памяти

Есть 2 типа кассет памяти: EEPROM и EPROM. В таблице показаны кассеты памяти, которые можно использовать с C200HX/HG/HE.

Тип памяти	Объем	Номер модели	Комментарии
EEPROM	4К слов	C200HW-ME04K	Кассету памяти EEPROM можно использовать для чтения и записи UM и IOM. Не требуется питания и данные будут сохраняться даже после удаления кассеты от ЦПУ
	8К слов	C200HW-ME08K	
	16К слов	C200HW-ME16K	
	32К слов	C200HW-ME32K	
EPROM	16 или 32 К слов	C200HS-MP16K	Микросхема EPROM не включена в кассету памяти и должна покупаться отдельно. 27256 эквивалент (ROM-JD-B): 16К 27512 эквивалент (ROM-KD-B): 32К

- Замечание**
1. Данные, сохраняемые в EEPROM, будут ненадежны после 50 000 операций записи.
  2. Для записи в кассету памяти EPROM пользуйтесь стандартным записывающим устройством. Перед установкой кассеты памяти EPROM в ЦПУ поставьте на нее EPROM. Кассета памяти EPROM потеряет информацию после убирания ее от ЦПУ.

### 2.4.1 Установочные параметры и начальные установки аппаратуры

Далее описаны установочные параметры и начальные установки аппаратуры, относящиеся к Кассете памяти.

#### Установки переключателя

На заводе-изготовителе переключатель 1 на кассете памяти = OFF. Перед установкой проверьте установки переключателя 1.

Кассета памяти	Установки перекл. 1	Функция
EEPROM	ON	Данные в Кассете памяти защищены от записи
	OFF	Данные в Кассете памяти не защищены от записи

Кассета памяти	Установки перекл. 1	Функция
EPROM	ON	27512 эквивалент (ROM-KD-B) EPROM (32К слов, время доступа 150 нс)
	OFF	27256 эквивалент (ROM-JD-B) EPROM (16К слов, время доступа 150 нс)

#### Флаги области SR и биты управления

SR 269..SR 273 содержат флаги и биты управления, относящиеся к содержимому и операциям Кассеты памяти. Подробности см.п.3.4.

#### 2.4.2 Запись/чтение содержимого UM

Для передачи содержимого UM на или с Кассеты памяти пользуйтесь следующими процедурами. (Для записи в кассету PROM требуется записывающее устройство. Подробности см. Руководство по работе с ПО).

**Замечание** В области UM находится релейно-контактная программа, фиксированная DM, дополнительная DM, комментарии входов/выходов, таблица входов/выходов и информация о распределении UM.

#### Запись содержимого UM на кассету памяти

Для записи в Кассету EEPROM пользуйтесь следующими процедурами.

- 1, 2, 3,... 1. Перед подачей питания на C200HX/HG/HE убедитесь, что переключатель 1 на кассете установлен в OFF.
2. Включите C200HX/HG/HE и запишите релейно-контактную программу или прочитайте готовую программу с диска данных.
3. Включите C200HX/HG/HE в режим PROGRAM.
4. С помощью управляющего компьютера с ПО или Консоли программирования включите SR 27000 (бит записи UM на кассету) в ON. Данные переписутся с ПК на Кассету памяти. После завершения записи SR 27000 автоматически сбросится в OFF.
5. Если Вы хотите защитить от записи данные на Кассете памяти, выключите ПК и установите переключатель 1 на кассете памяти в ON. Если данный переключатель уже в ON, данные на кассете будут сохраняться даже при включения в 1 SR 27000.

#### Чтение содержимого UM с Кассеты памяти

Есть 2 способа чтения с кассеты памяти: автоматическая передача при пуске ПК и однократная передача с помощью периферийного устройства.

(Функции автоматической записи на не Кассету памяти нет).

#### Автоматическая передача при пуске ПК.

- 1, 2, 3,... 1. Включить секцию 2 DIP переключателя на ЦПУ в ON.
2. Установите кассету памяти с информацией в C200HX/HG/HE.
3. Подайте питание на C200HX/HG/HE. Содержимое кассеты автоматически переписется в ЦПУ. Если данные не считались, появится сообщение об ошибке памяти.

#### Однократная передача с помощью периферийного устройства

- 1, 2, 3,... 1. Установите кассету памяти с информацией в C200HX/HG/HE.
2. Подайте питание на C200HX/HG/HE и переключите в режим PROGRAM.
3. Воспользуйтесь управляющим компьютером с ПО или Консолью программирования для установки бита SR 27001 (бит загрузки UM с кассеты) в ON. Данные будут считаны с Кассеты памяти на ПК. После завершения записи SR 27001 автоматически сбросится в OFF.

#### Сравнение содержимого UM с содержимым Кассеты памяти

- 1, 2, 3,... 1. Включите C200HX/HG/HE в режим PROGRAM.



2. Воспользуйтесь управляющим компьютером с ПО или Консолью программирования для установки бита SR 27002 (бит сравнения UM с кассетой) в ON. Будут сравниваться данные на Кассете памяти с памятью ПК. После завершения записи SR 27002 автоматически сбросится в 0.

3. Воспользуйтесь управляющим компьютером с ПО или Консолью программирования для проверки бита SR 27003 (флаг результата сравнения).

**Замечание** Если сравнение выполняется не в режиме PROGRAM, появится признак ошибки FAL90 и SR 27002 установится в ON. Хотя SR 27003 также будет в ON, сравнения не произойдет. Если сравнение проходило при неустановленной кассете памяти, SR 27003 установится в ON.

### 2.4.3 Запись/чтение содержимого IOM

Для передачи содержимого IOM на или с Кассеты памяти пользуйтесь следующими процедурами. (Для записи в кассету EPROM требуется записывающее устройство. Подробности см. Инструкцию по работе с ПО).

IOM включает области IR, SR, LR, HR, AR, текущие значения счетчиков и таймеров, DM 0000..DM 6143 и EM 0000..EM 6143.

При чтении/записи емкость кассеты памяти должна соответствовать емкости памяти ЦПУ. Требования к объему:

Запись IOM: объем памяти ЦПУ ≤ емкость кассеты памяти.

Чтение IOM: объем памяти ЦПУ = объем информации IOM на кассете памяти

В таблице показан объем Кассеты памяти, требуемый для сохранения одного или более банков EM.

Объем кассеты памяти	Количество банков EM
4 К слов	Нет (Кассету емкостью 4 К слов нельзя использовать для сохранения IOM)
8 К слов	Нет
16 К слов	1 банк (можно сохранить только банк 0 EM)
32 К слов	3 банка (можно сохранить банки EM 0..2)

Биты 08..15 в SR 273 указывают на число банков IOM, загруженных в кассету памяти.

Содержание SR 27308..SR 27315	Значение
00	Кассета памяти не установлена, или в Кассете нет данных IOM, или в Кассете нет данных EM
01	Кассета памяти содержит данные IOM, которые включают только банк 0. (Сохраняется в C200HG-CPU__-E)
04	Кассета памяти содержит данные IOM, которые включают банки 0..3. (Сохраняется в C200HX-CPU__-E)

### Запись содержимого IOM на Кассету памяти

Для записи содержимого IOM в Кассету EEPROM пользуйтесь следующими процедурами.

- 1, 2, 3,... 1. Перед подачей питания на C200HX/HG/HE убедитесь, что переключатель 1 на кассете установлен в OFF.
2. Включите C200HX/HG/HE в режим PROGRAM.
3. С помощью управляющего компьютера с ПО или Консоли программирования включите SR 27300 (бит записи IOM на кассету) в ON. Данные переписутся с ПК на Кассету памяти. После завершения записи SR 27300 автоматически сбросится в OFF.
4. Если Вы хотите защитить от записи данные на Кассете памяти, выключите ПК и установите переключатель 1 на кассете памяти в ON. Если данный

переключатель уже в ON, данные на кассете будут сохраняться даже при включении SR 27300 в ON.

### Чтение содержимого IOM с Кассеты памяти

Для чтения содержимого IOM с Кассеты памяти пользуйтесь следующими процедурами. Содержание протокола ошибок (DM 6000..DM 6030) нельзя прочитать с кассеты памяти.

**Замечание** Не существует команды автоматического чтения IOM с кассеты памяти.

1. Установите кассету с информацией в C200HX/HG/HE.
2. Подайте питание на C200HX/HG/HE и включите в режим PROGRAM.
3. С помощью управляющего компьютера с ПО или Консоли программирования включите SR 27301 (бит загрузки IOM с кассеты) в ON. Данные прочитаются с кассеты памяти на ПК. После передачи данных SR 27301 автоматически сбросится в OFF.

## 2.5 Работа без батареи

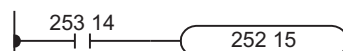
C200HX/HG/HE - ZE могут работать без батареи, пока соблюдаются следующие условия.

- Программа пользователя загружена в Кассету памяти (EPROM или EEPROM).
- Часы не используются. (Для внутренних часов нужна батарея).
- Установочные параметры должны быть настроены не генерировать ошибку батареи.
- Система разработана так, чтобы корректно работать при потерянных данные в области AR.
- Программа пользователя написана так, что SR25215 (бит выключения выходов в 0) не включается в 1. (Состояние этого бита будет нестабильно при отсутствии батареи).
- SR 25215 (Бит удержания состояния входов/выходов) и SR 25211 (Бит удержания принудительного состояния) настроены на очистку при пуске ПК. (Состояние этих битов будет нестабильно при отсутствии батареи).
- Секция 1 переключателя DIP на ЦПУ включен в OFF, а секция 1 включен в ON.

Используйте следующую процедуру.

- 1, 2, 3,...**
1. С помощью пакета поддержки выделите область UM. (Этот шаг необходим, когда область расширенных DM будет использоваться специальными Модулями или когда комментарии входов/ выходов должны быть сохранены в Модуле ЦПУ.
  2. Создайте программу пользователя и передайте ее в Модуль ЦПУ. Добавьте следующие команды в программу пользователя для обеспечения того, чтобы бит SR 25215 (бит выключения выходов в 0) не включается в 1.

**Замечание** Хотя SR 25215 будет сохранять предыдущее состояние при каждом отключении питания (при работе с батареей), его состояние будет нестабильно при работе без батареи. Следующую строку нужно добавить в программу для обеспечения того, чтобы он оставался в 0.



3. Сделайте следующие настройки установочных параметров.
  - a. Установите DM 6601 = 0000, чтобы IOM и принудительное состояние было сброшено при пуске.
  - b. Установите DM 6655 = 1x0x, чтобы ошибки батареи не обнаруживались. («x» указывает значения, которые можно задавать как 0, так и 1).
  - c. Установите DM 6600 DM 6602..DM 6654 как требуется.
4. Установите область фиксированных DM, включая параметры панели связи в DM 6144..DM 6599.
5. Установите область расширенных DM, если необходимо.

6. Подтвердите операцию.
7. Установите кассету памяти в Модуль ЦПУ и переключитесь в режим PROGRAM.
8. Включите SR 27000 = 1 для записи программы, фиксированных DM и установочных параметров в кассету памяти.

**Замечание** Когда используется Консоль программирования, данный бит включится в 1 автоматически после завершения записи. Если используется Пакет поддержки, бит будет установлен принудительно и состояние принудительной установки должно быть снято перед выключением этого бита в 0.

9. Включите в ON переключатель защиты от записи на кассете памяти.

### 2.6 Переключатель DIP ЦПУ

Шесть секций переключателя DIP управляют шестью рабочими параметрами ЦПУ.

Секция	Параметр	Положение	Функция
1	Защита памяти	ON	Нельзя производить запись в область UM <sup>1</sup> с периферийного устройства
		OFF	Можно производить запись в область UM <sup>1</sup> с периферийного устройства
2	Автоматическая пересылка содержимого кассеты памяти	ON	Содержимое Кассеты памяти будет автоматически передано в RAM при пуске
		OFF	Содержимое Кассеты памяти не будет автоматически передано при пуске
3	Язык сообщений	ON	Сообщения на программаторе пишутся по-английски
		OFF	Сообщения на программаторе пишутся на языке, загруженном в ROM (при японской версии - по-японски)
4	Параметры дополнительных инструкций	ON	Дополнительные инструкции заданы пользователем. Нормально ON при использовании управляющего компьютера для программирования/просмотра
		OFF	Дополнительные инструкции установлены по умолчанию
5	Параметры связи	ON	Стандартные параметры для Porta связи (включая Периферийный порт, когда подключен CQM1-CIF01. 1 стартовый бит, 7 битов данных, контроль на четность: четн, 2 стоповых бита, скорость обмена 9 600 бод)
		OFF	Связь не управляется параметрами по умолчанию
6	Режим расширенный TERMINAL	ON	Консоль программирования в режиме расширенного терминала (AR 0712 = ON)
		OFF	Консоль программирования в нормальном режиме (AR 0712 = OFF)

- Замечание**
1. В области UM находится релейно-контактная программа, фиксированная DM, дополнительная DM, комментарии входов/выходов, таблица входов/выходов и информация о распределении UM.
  2. При поставке все 6 секций установлены в OFF.



### **3. Области памяти**

*Для достижения эффективного и правильного управления требуются данные различного типа. Для облегчения управления этими данными в ПК есть различные области памяти для данных, каждая для выполнения своей особой функции. Эти области как правило, доступны для пользователя для программирования и называются областями данных. Другая область памяти - область УМ, где хранится программа пользователя. В данной главе эти области описаны отдельно и представлена информация, необходимая при работе с памятью. В данной главе описана также область TR, хотя она и не является в строгом смысле областью памяти.*

## 3.1 Введение

### 3.1.1 Общий обзор области данных

Подробная информация о данных, включая имя, размер и диапазон каждой области данных, обобщена в следующей таблице. К областям данных и памяти обращение производится по их акронимам (сокращенным названиям), например область AR, область SR и т. д.

Область данных	Размер	Диапазон	Комментарии
IR 1 Внутренние реле область 1	3 776 бит	IR 000.. IR 235	Подробности см. 3.1.2 и 3.3
SR1 Специальные реле область 1	312 бит	SR 236..SR 255	Подробности см. 3.1.2 и 3.4
SR2 Специальные реле область 2	704 бит	SR 256..SR 299	
IR2 Внутренние реле область 2	3 392 бит	IR 300..IR 511	Подробности см. 3.1.2 и 3.3
TR Область временных реле	8 бит	TR 00..TR 07	Служит для временного хранения и восстановления условий исполнения при программировании определенных типов ветвления релейно-контактной схемы
HR Область удерживающих реле	1 600 бит	HR 00..HR 99	Служит для хранения данных при отключении питания ПК
AR Область вспомогательных реле	448 бит	AR 00..AR 27	Содержит флаги и биты для специальных функций. Сохраняет состояние при отключенном питании
LR Область реле связи	1 024 бит	LR 00.. LR 63	Служит для обмена данными в системе связи PC LINK SYSTEM. Данные биты можно использовать в качестве рабочих, если PC LINK SYSTEM не используется.
ТС Область счетчиков/таймеров	512 счетчиков/таймеров	ТС 000..ТС 511	Служит для задания таймеров и счетчиков и доступа к флагам завершения, текущим и заданным значениям
Область DM	6 144 слов	DM 0000..DM 6143	Чтение/запись
	1 000 слов	DM 0000..DM 0999	Нормальные DM
	2 600 слов	DM 1000..DM 2599	Область специальных модулей
	3 400 слов	DM 2600..DM 5999	Нормальные DM
	31 слово	DM 6000..DM 6030	Протокол ошибок
	(44 слова)	DM 6100..DM 6143	Область проверки связи (резерв)
Область фиксированных DM	512 слов	DM 6144..DM 6599	Фиксированные DM (только чтение)
	56 слов	DM 6600..DM 6655	Установочные параметры ПК

Область данных	Размер	Диапазон	Комментарии
EM Область расширенной памяти данных	6 144 слов	EM 0000..EM 6143	Объем памяти EM зависит от модели ПК. Есть модели без EM, с 1 банком на 6 144 слова, или 3 банка на 6 144 слова. Аналогично DM, память EM доступна только в виде слов и она сохраняется при отключении питания.

#### Рабочие биты и слова

Если биты и слова в некоторой области данных не используются для специальных целей, их можно программировать произвольно для управления другими битами. Слова и биты, которые можно использовать для этого, называются рабочими словами и битами. Они описаны (по областям) в конце данной главы. Их применение описано в гл. 4.

#### Флаги и биты управления

В некоторых областях содержатся флаги и/или управляющие биты. Флаги и управляющие биты автоматически устанавливаются в 0 или 1 для указания определенного состояния операции. Хотя некоторые флаги могут быть установлены в 1 или в 0 пользователем, большинство флагов только читаются. Ими нельзя управлять непосредственно.

Управляющие биты - это биты, которые включаются в 1 или 0 пользователем для управления определенными аспектами отработки программы. Любой бит, имя которого дается с помощью бита слова, а не флага слова, является управляющим битом, например, бит перезапуска является управляющим битом.

### 3.1.2 Общие сведения об области IR/SR

При обозначении области данных для всех областей данных, кроме IR и SR, требуется акроним (сокращенное название). Хотя акронимы для областей IR и SR даются для ясного понимания текста, они не требуются и при программировании не вводятся.

Области IR и SR делятся на 2 секции по 256 слов; граница между этими секциями расположена в области SR между SR 255 и SR 256. Когда SR используется в качестве операнда команды, операнд не может переходить границу. Кроме того, базовые команды, которые обращаются к битам во второй секции (SR 25600..IT 51115), имеют большее время исполнения.

Область данных	Диапазон	Комментарии	
IR 1	Область 1 входов/выходов	IR 000..IR 029	Слова входов/выходов выделены панели ЦПУ и панелям расширения входов/выходов
	Область модулей группы-2 и модулей интерфейса B7A	IR 030..IR 049	Выделены модулям входов/выходов группы-2 и модулям интерфейса B7A 0..9
	Область SYSMAC BUS	IR 050..IR 099	Выделена ведомым панелям удаленных входов/выходов 0..4
	Область 1 модулей специальных входов/выходов	IR 100..IR 199	Выделена специальным модулям входов/выходов 0... 9
	Область модулей оптических входов/выходов и терминалов входов/выходов	IR 200..IR 231	Выделена модулям оптических входов/выходов и терминалам входов/выходов
	Рабочая область	IR 232..IR 235	Для использования в качестве рабочих бит в программе
SR 1	SR 23600.. SR 25507	Содержит системные часы, флаги, управляющие биты и информацию о состоянии контроллера	

### 3.2 Структура области данных

Область данных		Диапазон	Комментарии
SR 2		SR 256.. SR 299	Содержит флаги, управляющие биты и информация о состоянии контроллера. SR 290..SR 299 используются в качестве слов входов/выходов командой MCRO
IR 2	Область 2 входов/выходов	IR 300..IR 309	Слова входов/выходов выделены третьей панели расширения входов/выходов
	Рабочая область	IR 310..IR 329	Для использования в качестве рабочих битов в программе
	Область 2 модулей входов/выходов группы 2	IR 330..IR 341	Выделены модулям входов/выходов группы 2
	Рабочая область	IR 342..IR 349	Для использования в качестве рабочих битов в программе
	Область 2 модулей специальных входов/выходов	IR 350..IR 399	Не выделена
		IR 400..IR 459	Выделена модулям специальных входов/выходов A..F
Рабочая область	IR 460.. IR511	Для использования в качестве рабочих бит в программе	

- Замечание**
1. Подробности об области IR см. 3.3. Подробности об области SR см. 3.4.
  2. Биты в области IR 1 и области IR 2 можно использовать в программировании как рабочие биты, если они не задействованы на специальные цели.

### 3.2 Структура области данных

При назначении области данных для всех областей данных, кроме IR и SR, требуется акроним (сокращенное название). Хотя акронимы для областей IR и SR даются для ясного понимания текста, они не требуются и при программировании не вводятся. Любое обращение к данным без акронима понимается, что идет обращение к областям IR или SR. Поскольку адреса IR и SR идут последовательно, адреса бита или слова достаточно для различия этих двух областей.

Фактическое положение данных в области данных (кроме области TC) задается адресом. Адрес обозначает бит или слово в области данных, где расположен требуемый параметр. Область TC состоит из номеров TC, каждый из которых используется для конкретного таймера или счетчика, определенного в программе. Подробности о номерах TC см. 3.8, а информацию о применении таймеров см. 5.14.

Остальные области данных (например, IR, SR, HR, DM, AR и LR) состоят из слов, каждое из которых состоит из 16 битов, нумеруемых 00..15 справа налево. Слова IR 000..001 показаны далее с номерами битов. Здесь содержимое каждого слова показано как все нули. Бит 00 называется самым правым битом; бит 15 называется самым левым битом.

Термин самый младший бит часто используется для самого правого бита; термин самый старший бит часто используется для самого левого бита. Данные термины не используются в данном руководстве, поскольку слова часто разбиваются на 2 или более частей, каждая часть используется для различных параметров или операндов. Когда это сделано, самые правые биты слова могут фактически стать самыми старшими битами, т.е. самыми левыми битами в другом слове, когда они комбинируются с другими битами для формирования нового слова.

Номер бита	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
IR 000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IR 001	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Область DM доступна только в виде слов; Вы не можете указать отдельный бит внутри слова DM. Данные в областях IR, SR, HR, AR и LR доступны в виде слов и в виде битов, в зависимости от команды, в которой используются эти данные.

Для обращения к одной из этих областей в виде слова, все, что необходимо - это акроним (если требуется) и адрес слова из одной, двух, трех или четырех цифр. Для обращения к одной из этих областей в виде бита к адресу слова добавляется номер бита и получается единый 4- или 5-разрядный адрес. В следующей таблице показаны примеры. Две самых правых цифры обозначения бита должны быть в диапазоне 00..15. Один и тот же номер TC можно использовать для обозначения либо текущего значения таймера или счетчика, либо бита, который функционирует как флаг завершения таймера или счетчика. Это объясняется в 3.8.

Область	Обозначение слова	Обозначение бита
IR	000	00015 (самый левый бит в слове 000)
SR	252	25200 (самый правый бит в слове 252)
DM	DM 1250	Невозможно
TC	TC 215 (текущее значение)	TC 215 (флаг завершения)
LR	LR 12	LR 1200

#### Структура данных

Слово данных, вводимое в виде десятичного числа, хранится в виде двоично-десятичного значения; слово данных, введенное в 16-ричном виде хранится в двоичном виде. Каждые 4 бита слова представляют одну цифру, 16-ричную или десятичную. Таким образом, одно слово данных содержит 4 цифры, нумеруемые справа налево. Соответствие номеров цифр и битов показаны ниже.

Номер цифры	3				2				1				0			
Номер бита	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Содержание	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

При обращении к целому слову цифра под номером 0 называется самой правой цифрой; под номером 3 - самой левой цифрой.

При вводе данных в область данных, они должны вводиться в должной форме для конкретных целей. Это не составляет проблемы при обозначении индивидуальных битов, которые просто включаются в 1 или 0. При вводе данных в виде слова важно водить число либо в десятичном, либо в 16-ричном виде, в зависимости от требований команды, для которой используется это слово. В гл. 5 указаны формы, требуемые командой.

#### Преобразование различных форм данных

Двоичные и 16-ричные значения можно легко преобразовывать друг в друга, поскольку каждые 4 бита двоичной формы эквивалентны одной цифре 16-ричного числа. Двоичное число 010111101011111 преобразуется в 16-ричное с учетом каждой четверки битов в порядке справа налево. Двоичное 1111 - это 16-ричное F;

Двоичное 0101 - это 16-ричное 5; 16-ричный эквивалент данного числа = 5F5F, или 24 415 (163 x 5 + 162 x 15 + 16 x 5 + 15).

Двоичное и двоично-десятичное значение легко преобразовать друг в друга. В данном случае, каждая двоично-десятичная цифра (т.е. каждая группа четырех двоично-десятичных бит) эквивалентна соответствующей десятичной цифре.

Двоично-десятичные биты 0101011101010111 преобразуются в десятичное значение с учетом каждой четверки битов в порядке справа налево. Двоичное 0101 - это десятичное 5; Двоичное 0111 - это десятичное 7. Десятичный эквивалент данного числа = 5 757.

Обратите внимание, что это не то же самое значение, что 16-ричный эквивалент 0101011101010111, который равен 5 757 16-ричных, или 22 359 десятичных (163 x 5 + 162 x 7 + 16 x 5 + 7).

Поскольку числовой эквивалент каждой четырех двоично-десятичных битов должен быть десятичным, любая комбинация из 4 бит свыше 9 нельзя использовать, например, комбинация 1011 не разрешена, поскольку это эквивалент 11, что нельзя выразить одним десятичной цифрой. Комбинация двоичных битов 1011 разрешена в 16-ричном числе, это эквивалент цифры С.

Есть команды, предназначенные для преобразования данных между двоично-десятичным и двоичным представлением. Подробности см. 5.18. Таблица двоичных эквивалентов 16-ричным и десятичным числам приведены в приложениях.

#### Десятичные точки

Десятичные точки используются только в таймерах. Самая младшая цифра представляет собой десятую долю секунды. Все математические команды оперируют только с целыми числами.

#### Двоичные значения со знаком и без знака

В данном разделе объясняются форматы двоичные со знаком и без знака. Многие команды могут использовать двоичные числа и со знаком, и без знака, некоторые только со знаком (CPS, CPSL, DBS, DBSL, MBS, MBSL).

#### Двоичные без знака

Формат двоичный без знака - это стандартный формат ПК OMRON. Данные в настоящей инструкции являются без знака (если особо не оговорены другие форматы). Двоичные значения без знака всегда положительны и имеют диапазон значений 0 (#0000)... 65 535 (#FFFF). 8- разрядные числа имеют диапазон значений 0 (#0000 0000)... 4 294 967 295 (#FFFF FFFF).

Значение цифры	16 <sup>3</sup>				16 <sup>2</sup>				16 <sup>1</sup>				16 <sup>0</sup>			
Номер бита	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Содержание	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

#### Двоичные со знаком

Двоичное число со знаком может иметь положительное либо отрицательное значение. Знак указывается состоянием бита 15. Если бит 15 = 0, число положительно, если бит 15 = 1, число отрицательно.

Значения положительных чисел: 0 (#0000)... 32 767 (#7FFF).

Значения отрицательных чисел: -32 768 (#8000).. -1 (#FFFF).

Индикатор знака																
Значение цифры	16 <sup>3</sup>				16 <sup>2</sup>				16 <sup>1</sup>				16 <sup>0</sup>			
Номер бита	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Содержание	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Значения 8-разрядных положительных чисел:

0 (#0000 0000)... 2 147 483 647 (#7FFF FFFF).

Значения 8-разрядных отрицательных чисел:

2 147 483 648 (#8000 0000)..-1 (#FFFF FFFF).

В следующей таблице показаны соответствующие значения десятичное, 16-ричное и 16-ричное 32-битовое.

Десятичные	16-ричные 16 бит	16-ричные 32 бита
2 147 483 647	-	7FFF FFFF
2 147 483 646	-	7FFF FFFE
..	..	..
32 768	-	0000 8000
32 767	7FFF	0000 7FFF
32 766	7FFE	0000 7FFE

Десятичные	16-ричные 16 бит	16-ричные 32 бита
2	0002	0000 0002
1	0001	0000 0001
0	0000	0000 0000
-1	FFFF	FFFF FFFF
-2	FFFE	FFFF FFFE
..	..	..
-32 767	8001	FFFF 8001
-32 768	8000	FFFF 8000
-32 769	-	FFFF 7FFF
..	..	..
-2 147 483 647	-	8000 0001
-2 147 483 648	-	8000 0000

**Преобразование десятичных чисел в двоичные со знаком**

Положительное двоичное число со знаком идентично двоичному числу без знака и его можно преобразовать командой BIN(100). Далее показана процедура преобразования отрицательного десятичного числа диапазона -32 768..-1 в двоичное число со знаком. В данном примере -12345 преобразуется в CFC7.

1. Сначала берется абсолютное значение (12345) и преобразуется в двоичное без знака.

Номер бита	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Содержание	0	0	1	1	0	0	0	0	0	0	1	1	1	0	0	1

2. Находится дополнение:

Номер бита	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Содержание	1	1	0	0	1	1	1	1	1	1	0	0	0	1	1	0

3. Наконец, добавляется 1:

Номер бита	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Содержание	1	1	0	0	1	1	1	1	1	1	0	0	0	1	1	1

Для преобразования двоичного числа со знаком в десятичное выполните операции в обратном порядке.

**3.3 Область IR (внутренние реле)**

Область IR служит и в качестве области, управляемой точками входа/выхода, и в качестве рабочих бит для обработки и хранения данных. Эта область доступна как в виде слов, так и в виде битов. В C200NX/HG/HE область IR охватывает слова IR 000..IR 235 (область IR 1) и IR 300..IR 511 (область IR 2). Базовые команды имеют более долгое время исполнения при обращении к области 2.

Слова в области IR, которые используются для управления точками входов/выходов, называются словами входов/выходов. Биты в словах входов/выходов называются битами входов/выходов. Биты области входов/выходов, которые не привязаны к точкам входов/выходов, можно использовать в качестве рабочих битов. Рабочие биты области IR сбрасываются при отключении питания или прекращении отработки программы ПК.

Область данных	Диапазон	
IR 1	Область 1 входов/выходов	IR 000..IR 029
	Область модулей группы-2 и модулей интерфейса B7A	IR 030..IR 049
	Область SYSMAC BUS	IR 050..IR 099
	Область 1 модулей специальных входов/выходов	IR 100..IR 199

Область данных		Диапазон
	Область модулей оптических входов/выходов и терминалов входов/выходов	IR 200..IR 231
	Рабочая область	IR 232..IR 235
IR 2	Область 2 входов/выходов	IR 300..IR 309
	Рабочая область	IR 310..IR 329
	Область 2 модулей входов/выходов группы 2	IR 330..IR 341
	Рабочая область	IR 342..IR 349
	Область 2 модулей специальных входов/выходов	IR 350..IR 399
		IR 400..IR 459
Рабочая область	IR 460.. IR511	

#### Слова входов/выходов

Если Модуль поставляет входной сигнал на ПК, бит, привязанный к нему, называется входным битом. Если Модуль посылает выходной сигнал от ПК, бит, привязанный к нему, называется выходным битом. Для включения выхода в 1 нужно установить в 1 бит, привязанный к нему. Когда вход включается в 1, входной бит, привязанный к нему, также включается в 1. Это свойство можно использовать в программе для опроса состояния входных битов и управления состоянием выходных битов.

#### Применение входных битов

Входные биты можно использовать для прямого ввода внешних сигналов в ПК и при программировании могут использоваться в любом порядке. Каждый входной бит можно использовать в стольких командах, сколько необходимо для достижения эффективного и правильного управления. Их нельзя использовать в командах, которые управляют состоянием битов, например, OUTPUT, DIFFERENTIATION UP и KEEP.

#### Применение выходных битов

Входные биты используются для вывода результатов исполнения программы и при программировании могут использоваться в любом порядке. Поскольку выходы обновляются только раз за цикл (т.е. один раз за исполнение программы), выходной бит можно использовать только в одной команде, которые управляют состоянием битов, включая OUT, KEEP, DIFU, DIFD, SFT. Если выходной бит используется более, чем в одной такой команде, с ПК будет выдаваться состояние, установленное последней встреченной командой.

Пример использования выходного бита в двух командах, управляющих битами, см. 5.15.1.

#### Распределение слов в панелях

Слова входов/выходов выделены панели ЦПУ и панелям расширения входов/выходов согласно позициям платомест. Одно слово входов/выходов выделено одному платоместу, как показано в следующей таблице. Поскольку каждое платоместо привязано только к одному слову входов/выходов, панель на 3 платоместа использует только первые три слова, панель на 5 платомест использует только первые 5 слов, панель на 8 платомест использует только первые 8 слов. Слова, выделенные неиспользованным или несуществующим платоместам, можно использовать в качестве рабочих битов.

Панель	Левая сторона панели					Правая сторона панели (на 10 платомест)				
	Слот									
	1	2	3	4	5	6	7	8	9	10
CPU	IR 000	IR 001	IR 002	IR 003	IR 004	IR 005	IR 006	IR 007	IR 008	IR 009
Расширение 1	IR 010	IR 011	IR 012	IR 013	IR 014	IR 015	IR 016	IR 017	IR 018	IR 019
Расширение 2	IR 020	IR 021	IR 022	IR 023	IR 024	IR 025	IR 026	IR 027	IR 028	IR 029
Расширение 3	IR 300	IR 301	IR 302	IR 303	IR 304	IR 305	IR 306	IR 307	IR 308	IR 309

**Неиспользованные слова**

Любые слова, выделенные модулю, который их не использует, можно использовать в качестве рабочих слов или битов. Модули, которые не используют слова, выделенные платоместам, на которые они установлены, включают модули связи (например, модули HOST LINK, PC LINK, SYSMAC NET и т. д.), ведущие модули удаленных входов/выходов, специальные модули входов/выходов, модули группы 2, модули интерфейса В7А группы 2 и модули дополнительного питания.

**Распределение памяти для модулей специальных входов/выходов и ведомых панелей**

В большинстве С200НХ/НГ/НЕ можно смонтировать до 16 модулей специальных входов/выходов (на любые платоместа) на панели ЦПУ или панели расширения входов/выходов. (Ограниченное число модулей специальных входов/выходов можно также установить на ведомых панелях удаленных входов/выходов). Каждому модулю специальных входов/выходов выделены 10 слов, определяемых номером модулей (0..F). На С200НЕ-CPU\_\_-E(ZE) и С200НГ/НХ-CPU3\_-E/4\_-E(ZE) можно установить до 10 модулей специальных входов/выходов. Каждому модулю специальных входов/выходов выделены 10 слов, определяемых номером модулей (0..9).

Номер модуля	Слова входов/выходов	Ограничения ПК
0	IR 100..IR 109	Нет
1	IR 110..IR 119	
2	IR 120..IR 129	
3	IR 130..IR 139	
4	IR 140..IR 149	
5	IR 150..IR 159	
6	IR 160..IR 169	
7	IR 170..IR 179	
8	IR 180..IR 189	
9	IR 190..IR 199	
A	IR 400..IR 409	Нет у С200НЕ-CPU__-E(ZE) и С200НГ/НХ-CPU3_-E(ZE) /4_-E(ZE)
B	IR 410..IR 419	
C	IR 420..IR 429	
D	IR 430..IR 439	
E	IR 440..IR 449	
F	IR 450..IR 459	

**Замечание** Слова входов/выходов, которые не выделены модулям специальных входов/выходов, можно использовать в качестве рабочих бит.

Можно использовать до 5 ведомых панелей, когда используются один или два ведущих. Слова области IR выделяются ведомым панелям согласно номеру модулей, как показано в таблице.

Номер модуля	Слова входов/выходов
0	IR 050..IR 059
1	IR 060..IR 069
2	IR 070..IR 079
3	IR 080..IR 089
4	IR 090..IR 099

Можно использовать ведомую панель удаленных входов/выходов С500-RT001/002-(P)V1, но она требует 20 слов входов/выходов, а не 10, и, следовательно, занимает слова входов/выходов, выделенных для двух ведомых панелей С200Н, слова, выделенные номеру модулей на панели и номеру следующего модулей. При использовании ЦПУ С200НХ/НГ/НЕ не задавайте номера модулей на ведомой панели С500 = 4, поскольку нет модулей номер 5. При ведомой панели С500 слова входов/выходов выделяются

только установленным модулям, слева направо, а не платоместам. как на панелях C200HX/HG/HE.

**Выделение памяти модулям оптических входов/выходов и терминалам входов/выходов**

Слова входов/выходов IR 200..IR 231 выделяются модулям оптических входов/выходов и терминалам входов/выходов согласно номерам модулей. Слово входов/выходов, выделенное каждому модулю = 200 + n, где n - номер модулей, установленный на модуле.

**Выделение памяти удаленным ведущим входов/выходов и модулям связи**

Удаленные ведущие входы/выходы и модули связи HOST LINK не используют слова входов/выходов, и модули связи ПК используют область LR, так что слова, выделяемые платоместам, на которых устанавливаются данные модули, доступны в качестве рабочих слов.

**Выделение битов для модулей входов/выходов**

Модуль входов/выходов может потребовать от 8 до 16 бит, в зависимости от модели. У большинства модулей входов/выходов биты, не задействованные в качестве входа или выхода, доступны в качестве рабочих битов. модули транзисторных выходов C200H-OD213 и C200H-OD411, а также Модуль симисторных выходов C200H-OA221 используют бит 08 как флаг срабатывания предохранителя. Модуль транзисторных выходов C200H-OD214 использует биты 08..11 как флаг предупреждения о неисправности. Биты 08..15 любого слова, выделенного этим модулям, можно использовать в качестве рабочих бит.

**Выделение битов для модулей входов с прерываниями**

Модуль входов с прерываниями использует 8 бит первого слова входов/выходов, выделенного платоместу панели ЦПУ. (Модуль входов с прерываниями будет работать как Модуль нормальных входов при установке на панели расширения входов/выходов). Остальные 24 бита, выделенные платоместу на панели ЦПУ, можно использовать в качестве рабочих битов.

**Выделение памяти для Модулей группы 2 и Модулей интерфейса В7А**

Модулям входов/выходов группы 2 высокой плотности и модулям интерфейса В7А выделены слова IR 030..IR 049 согласно номерам входов/выходов, установленным на них и они не используют слова, выделенные платоместам, на которых они установлены. Для модулей на 32 точки каждому модулю выделено 2 слова. Для модулей на 64 точки каждому модулю выделено 4 слова. Слова, выделяемые каждому номеру входов/выходов показаны в следующей таблице. Слова или части слов, не используемые для входов/выходов, можно использовать в качестве рабочих битов.

Модули на 32 точки		Модули на 64 точки	
№ модуля	Слова	№ модуля	Слова
0	IR 30..IR 31	0	IR 30..IR 33
1	IR 32..IR 33	1	IR 32..IR 35
2	IR 34..IR 35	2	IR 34..IR 37
3	IR 36..IR 37	3	IR 36..IR 39
4	IR 38..IR 39	4	IR 38..IR 41
5	IR 40..IR 41	5	IR 40..IR 43
6	IR 42..IR 43	6	IR 42..IR 45
7	IR 44..IR 45	7	IR 44..IR 47
8	IR 46..IR 47	8	IR 46..IR 49
9	IR 48..IR 49	9	Нельзя использовать

При установке номеров входов/выходов на модулях входов/выходов высокой плотности и модулях интерфейса В7А обеспечьте, чтобы одинаковые слова не были выделены более, чем одному модулю. Например, если номер входов/выходов 0 выделен модулю на 64 точки, номер 1 нельзя использовать для другого модулей системы.

Модули входов/выходов высокой плотности группы 2 и модули интерфейса В7А не рассматриваются, как модули специальных входов/выходов и не влияют на выделение слов модулям специальных входов/выходов в системе, независимо от используемого количества.

Слова, выделенные для модулей входов/выходов высокой плотности группы 2, соответствуют разъемам модулей, как показано в следующей таблице.

Модуль	Слово	Разъем/ряд
Модули на 32 точки	Первое	Ряд А
	Второе	Ряд В
Модули на 64 точки	Первое	CN1, Ряд А
	Второе	CN1, Ряд В
	Третье	CN2, Ряд А
	Четвертое	CN2, Ряд В

**Замечание** Модули входов/выходов группы 2 и модули интерфейса В7А нельзя устанавливать на ведомых модулях.

### 3.4 Область SR (специальных реле)

Область SR содержит флаги и биты управления, используемые для просмотра работы ПК, доступа к часам, и сигнализации ошибок. Диапазон адресов слов области SR: 236..299, адресов битов 23600..29915.

Область SR делится на 2 секции. Первая секция оканчивается на SR 255, а вторая начинается на SR 256. Когда слово SR служит как операнд команды, операнд должен переходить эту границу. Базовые команды, которые адресуются к области 2 SR, имеют более долгое время исполнения.

Область	Диапазон
SR, область 1	SR 23600..SR 25507
SR, область 2	SR 25600..SR 29915

В следующей таблице перечисляются функции флагов и битов управления области SR. Большинство этих битов описаны более подробно в таблице. Описания приведены по порядку номеров битов, за исключением того, что биты SYSTEM LINK сгруппированы вместе.

Если не оговорено особо, флаги находятся в состоянии 0 до возникновения заданных условий, когда они включаются в 1. Биты перезапуска обычно установлены в 0, но когда пользователь переключает один из них сначала в 1, затем в 0, указанный Модуль связи перезапускается. Другие биты управления установлены в 0 до тех пор, пока не будут установлены пользователем.

Не все слова и биты SR могут писаться пользователем. Обязательно проверьте функцию бита или слова перед тем, как использовать его в программе.

Слово (слова)	Бит (биты)	Функция
236	00..07	Область вывода состояния цепи для рабочего уровня 0 системы SYSMAC NET
	08..15	Область вывода состояния цепи для рабочего уровня 1 системы SYSMAC NET
237	00..07	Область вывода кода завершения для рабочего уровня 0 после выполнения команд SEND/RCV системы SYSMAC LINK/SYSMAC NET
	08..15	Область вывода кода завершения для рабочего уровня 1 после выполнения команд SEND/RCV системы SYSMAC LINK/SYSMAC NET
238..241	00..15	Область вывода состояния связи для рабочего уровня 0 системы связи SYSMAC NET / SYSMAC NET
242..245	00..15	Область вывода состояния связи для рабочего уровня 1 системы связи SYSMAC NET / SYSMAC NET

### 3.3 Область IR (внутренние реле)

Слово (слова)	Бит (биты)	Функция	
246	00..15	Не используется	
247..248	00..07	Флаги работы модулей PC LINK для модулей 16..31 или состояния связи для рабочего уровня 1	
	08..15	Флаги ошибок модулей PC LINK для модулей 16..31 или состояния связи для рабочего уровня 1	
249..250	00..07	Флаги работы модулей PC LINK для модулей 00..15 или состояния связи для рабочего уровня 0	
	08..15	Флаги ошибок модулей PC LINK для модулей 00..15 или состояния связи для рабочего уровня 0	
251*	00	Бит чтения ошибок удаленных входов/выходов	
	01..02	Не используется	
	03	Флаг ошибки удаленных входов/выходов	
	04..06	Номер ведомой панели и номер модулей удаленных входов/выходов, оптических входов/выходов или терминалов входов/выходов с ошибкой	
	07	Не используется	
	08..15	Номер ведущего модулей и слова, выделенного модулю удаленных входов/выходов, оптических входов/выходов или терминалу входов/выходов с ошибкой (16-ричное значение)	
252	00	Флаг ошибки SEND/RECV для рабочего уровня 0 системы SYSMAC LINK или SYSMAC NET	
	01	Флаг разрешения SEND/RECV для рабочего уровня 0 системы SYSMAC LINK или SYSMAC NET	
	02	Флаг работы рабочего уровня 0 системы связи	
	03	Флаг ошибки SEND/RECV для рабочего уровня 1 системы SYSMAC LINK или SYSMAC NET	
	04	Флаг разрешения SEND/RECV для рабочего уровня 1 системы SYSMAC LINK или SYSMAC NET	
	05	Флаг рабочего уровня 1 системы связи	
	06	Флаг ошибки связи модулей HOST LINK, установленного на панели (уровень 1)	
	07	Бит перезапуска модулей HOST LINK, установленного на панели (уровень 1)	
	08	Флаг ошибки порта RS-232C	
	09	Флаг перезапуска порта RS-232C	
	10	Бит очистки установочных параметров ПК	
	11	Бит удержания принудительного состояния	
	12	Бит управления сохранением данных	
	13	Бит перезапуска модулей HOST LINK, установленного на панели (уровень 0)	
	14	Не используется	
	15	Бит сброс выходов в 0	
	253	00..07	Область выдачи номера FAL (см. информацию об ошибках)
		08	Флаг падения напряжения аккумулятора
09		Флаг ошибки времени цикла	
10		Флаг ошибки проверки входов/выходов	
11		Флаг ошибки связи модулей HOST LINK, установленного на панели (уровень 0)	
12		Флаг ошибки удаленных входов/выходов	
13		Флаг всегда 1	
14		Флаг всегда 0	
	15	Флаг первого цикла	



### 3.3 Область IR (внутренние реле)

Слово (слова)	Бит (биты)	Функция	
254	00	Бит импульса каждую 1 мин	
	01	Бит импульса каждые 0.02 с	
	02	Флаг отрицательного значения (N)	
	03	Флаг исполнения MTR	
	04	Флаг переполнения + (для расчетов с двоичными со знаком)	
	05	Флаг переполнения - (для расчетов с двоичными со знаком)	
	06	Флаг окончания просмотра фронтов	
	07	Флаг исполнения команды STEP	
	08	Флаг исполнения команды HKY	
	09	Флаг исполнения команды 7SEG	
	10	Флаг исполнения команды DSW	
	11	Флаг ошибки модулей входных прерываний	
	12	Флаг первого цикла	
	13	Флаг ошибки программы прерываний	
	14	Флаг ошибки группы 2	
15	Флаг ошибки специальных модулей (специальных входов/выходов, PC LINK, HOST LINK, удаленных ведущих)		
255	00	Бит импульса каждые 0.1 с	
	01	Бит импульса каждые 0.2 с	
	02	Бит импульса каждые 1 с	
	03	Флаг ошибки исполнения команды (ER)	Данные флаги включаются в 0 после выполнения команды END, так что их состояние нельзя просмотреть с программатора  Таблицу с командами, которые влияют на эти флаги, см. приложение С
	04	Флаг переноса (CY)	
	05	Флаг Больше чем (GR)	
	06	Флаг равно (EQ)	
	07	Флаг меньше чем (LE)	
08..15	Зарезервированы системой для битов TR		
256..261	00..15	Зарезервированы системой	
262	00..15	Время исполнения самой длинной подпрограммы прерываний (в 0.1 мс)	
263	00..15	Количество программ прерываний с большим временем исполнения (8000..8255) (бит 15 - флаг прерывания)	
264	00..03	Код ошибок порта RS-232C 0: нет ошибки 1: ошибка четности 2: ошибка формата 3: ошибка переполнения 4: ошибка контрольной суммы кадра 5: ошибка времени 6: ошибка контрольной суммы 7: ошибка команды	
	04	Ошибка связи порта RS-232C	
	05	Флаг готовности посылки порта RS-232C	
	06	Флаг завершения приема порта RS-232C	
	07	Флаг переполнения приема порта RS-232C	

### 3.3 Область IR (внутренние реле)

Слово (слова)	Бит (биты)	Функция	
	08..11	Код ошибок периферийного порта 0: нет ошибки 1: ошибка четности 2: ошибка формата 3: ошибка переполнения 4: ошибка контрольной суммы кадра 5: ошибка времени 6: ошибка контрольной суммы 7: ошибка команды	
	12	Ошибка связи периферийного порта	
	13	Флаг готовности посылки периферийного порта	
	14	Флаг завершения приема периферийного порта	
	15	Флаг переполнения приема периферийного порта	
265	00..15	NT LINK Режим (1:N) Биты 00..07: флаги связи с ПТ для модулей 0..7 Биты 08..15: флаги приоритета регистрации ПТ для модулей 0..7 Режим RS-232C Биты 00..15: Счетчик приема порта RS-232C	
266	00..15	Счетчик приема периферийного порта в режиме RS-232C	
267	00..04	Зарезервировано системой (пользователю не доступно)	
	05	Флаг готовности передачи HOST LINK, уровень 0	
	06..12	Зарезервировано системой (пользователю не доступно)	
	13	Флаг готовности передачи HOST LINK, уровень 1	
	14..15	Не используется	
268	00..15	Информация об ошибке панели связи	
269	00..07	Содержание кассеты памяти 00: ничего 01: UM 02: IOM 03: HIS	
	08..10	Объем кассеты памяти 0: нет кассеты 2: 4 или 8 К слов 3: 16 К слов 4: 32 К слов	
	11..13	Зарезервировано системой (пользователю не доступно)	
	14	Флаг защиты от записи кассеты памяти EEPROM или кассеты памяти с EPROM	
	15	Флаг кассеты памяти	
270	00	Бит сохранить UM на кассету	Данные передаются при включении этого бита в 1 в режиме PROGRAM Если эти биты включить в 1 в режимах RUN или MONITOR, появится признак нефатальной ошибки.
	01	Бит Загрузить с кассеты в UM	
	02	Бит сравнения UM с кассетой	
	03	Результаты сравнения 0: содержимое одинаково 1: содержимое различается или сравнение невозможно	
	04..10	Не используется	

### 3.3 Область IR (внутренние реле)

Слово (слова)	Бит (биты)	Функция	
	11	Флаг ошибки передачи: Передача таблицы SYSMAC NET в UM при активной линии связи	Данные не будут передаваться с UM на кассету памяти при ошибке (за исключением Ошибки контрольной суммы панели). Подробная информация об ошибках контрольной суммы в кассете памяти не будет выдана в SR 272, поскольку такая информация не нужна. Повторите передачу, если SR 27015 = 1.
	12	Флаг ошибки передачи: не в режиме PROGRAM	
	13	Флаг ошибки передачи: Только чтение	
	14	Флаг ошибки передачи: Недостаточно памяти или не UM	
	15	Флаг ошибки передачи: Ошибка контрольной суммы панели	
271	00..07	Размер релейно-контактной программы, записанной на кассету памяти: Файл только релейно-контактной программы: 04: 4 К слов 08: 8 К слов 12: 12 К слов .. 32: 32 К слов 00: Нет релейно-контактной программы, или загружен файл не с релейно-контактной программой.	
	08..15	Размер релейно-контактной программы в ЦПУ (Значения те же, что и для битов 00..07)	
272	00..10	Не используется	
	11	Флаг ошибки памяти: Ошибка контрольной суммы установочных параметров ПК	
	12	Флаг ошибки памяти: Ошибка контрольной суммы релейно-контактной программы	
	13	Флаг ошибки памяти: Ошибка контрольной суммы области вектора изменения команды	
	14	Флаг ошибки памяти: Выход кассеты из режима ON-LINE	
	15	Флаг ошибки памяти: Ошибка автозагрузки	
273	00	Бит Сохранить IOM на кассету	Данные передаются в кассету, когда бит = 1 в режиме PROGRAM. Бит автоматически сбрасывается в 0. При включении бита в 1 в любом другом режиме появляется признак ошибки.
	01	Бит загрузить IOM с кассеты	
	02	Установите этот бит в 0	
	03..07	Не используется	
	08..11	Содержит номер банка EM, когда в кассете памяти хранятся данные IOM	
	12	Флаг ошибки передачи: не в режиме PROGRAM	Данные не будут передаваться с IOM на кассету памяти при ошибке (за исключением Ошибки только чтения)
	13	Флаг ошибки передачи: Только чтение	
	14	Флаг ошибки передачи: Недостаточно памяти или не IOM	
	15	Всегда 0	
274	00	Флаг перезапуска модулей #0 специальных входов/выходов	Эти флаги будут установлены в 1 во время перезапуска. Эти флаги не будут установлены в 1 на ведомой панели.
	01	Флаг перезапуска модулей #1 специальных входов/выходов	
	02	Флаг перезапуска модулей #2 специальных входов/выходов	

### 3.3 Область IR (внутренние реле)

Слово (слова)	Бит (биты)	Функция	
	03	Флаг перезапуска модулей #3 специальных входов/выходов	
	04	Флаг перезапуска модулей #4 специальных входов/выходов	
	05	Флаг перезапуска модулей #5 специальных входов/выходов	
	06	Флаг перезапуска модулей #6 специальных входов/выходов	
	07	Флаг перезапуска модулей #7 специальных входов/выходов	
	08	Флаг перезапуска модулей #8 специальных входов/выходов	
	09	Флаг перезапуска модулей #9 специальных входов/выходов	
	10	Флаг перезапуска модулей #A специальных входов/выходов	
	11	Флаг перезапуска модулей #B специальных входов/выходов	
	12	Флаг перезапуска модулей #C специальных входов/выходов	
	13	Флаг перезапуска модулей #D специальных входов/выходов	
	14	Флаг перезапуска модулей #E специальных входов/выходов	
	15	Флаг перезапуска модулей #15 специальных входов/выходов	
	275	00	Ошибка установочных параметров ПК (DM 6600..DM 6605)
01		Ошибка установочных параметров ПК (DM 6613..DM 6623)	
02		Ошибка установочных параметров ПК (DM 6635..DM 6655)	
03		Не используется	
04		Флаг смены установочных параметров RS-232C	
05		Не используется	
06..07		Зарезервировано системой (недоступно пользователю)	
08..15		Не используется	
276	00..07	Минуты (00..59)	Индикация текущего времени в двоично-десятичном виде
	08..15	Часы (00..23)	
277..279	00..15	Используется для отображения состояния клавиатуры (см. 7.2.18)	
280	00..15	Флаги ошибки модулей входов/выходов группы 2 0..F (AR 0205..AR 0214 также выполняют функцию флагов ошибок для модулей 0...9)	
281	00..15	Биты перезапуска специальных модулей входов/выходов 0..F (модули 0..9 можно также запустить повторно битами перезапуска специальных модулей входов/выходов AR 0100..AR 0109)	
282	00..15	Флаги ошибки модулей специальных входов/выходов 0..F (AR 0000..AR 0009 также выполняют функцию флагов ошибок для модулей 0...9)	
283..286	00..15	Область просмотра панели связи	
287..288	00..15	Область параметров прерываний панели связи	
289	00..07	Область общего просмотра панели связи	
	08	Флаг исполнения команды порта А панели связи	
	09..10	Используется командами порта А панели связи	
	11	Бит прекращения команды порта А панели связи	

Слово (слова)	Бит (биты)	Функция
	12	Флаг исполнения команды порта В панели связи
	13..14	Используется командами порта В панели связи
	15	Бит прекращения команды порта В панели связи
290..293	00..15	Входы области MCRO
294..297	00..15	Выходы области MCRO
298..299	00..15	Зарезервировано системой (пользователю недоступно)

**Замечание** (\*) - для слова возможен режим записи.

### 3.4.1 Система SYSMAC NET/SYSMAC LINK

#### Состояние цепи

SR 236 показывает состояние цепи локального узла для системы SYSMAC NET, как показано далее

	Уровень 0	Уровень 1	Состояние/значение
SR 236	07	15	1
	06	14	1
	05	13	Напряжение центрального питания 0: подключено 1: не подключено
	04	12	1
	03	11	Состояние цепи: 11: нормальная цепь 10: цепь "вниз" 01: цепь "вверх" 00: ошибка цепи
	02	10	
	01	9	Состояние приема 0: прием разрешен 1: прием не разрешен
	00	8	1

#### Коды завершения

SR 23700..SR 23707 показывают коды завершения операций SEND/RCV рабочего уровня 0, а SR 23708..SR 23715 показывают коды завершения операций SEND/RCV рабочего уровня 1. Коды завершения приведены в следующих таблицах.

#### SYSMAC LINK

Код	Параметр	Значение
00	Нормальное завершение	Обработка завершилась нормально
01	Ошибка параметра	Параметры команды связи не в допустимом диапазоне
02	Нельзя посылать	Модуль сброшен при обработке команды или локального узла нет в сети
03	Адресата нет в сети	Узла адресата нет в сети
04	Ошибка "занято"	Узел адресата обрабатывает данные и не может принять команду
05	Время ответа истекло	Превышено время ожидания ответа
06	Ошибка ответа	Ошибка в ответе, принятом от узла адресата
07	Ошибка контроллера связи	Произошла ошибка контроллера связи
08	Ошибка установочных параметров	Ошибка в задании адреса узла
09	Ошибка ПК	Произошла ошибка в ЦПУ узла адресата

**SYSMAC NET**

Код	Параметр	Значение
00	Нормальное завершение	Обработка завершилась нормально
01	Ошибка параметра	Параметры команды связи не в допустимом диапазоне
02	Ошибка маршрутизации	Ошибка в таблице маршрутизации при связи с удаленной сетью
03	Ошибка "занято"	Узел адресата обрабатывает данные и не может принять запрос
04	Ошибка посылки (потеряна метка)	Не принята метка с линейного сервера
05	Ошибка цепи	Ошибка в цепи связи
06	Нет ответа	Узла адресата нет или превышено время ожидания ответа
07	Ошибка ответа	Ошибка в формате ответа

**Флаги состояния связи DATA LINK**

SR 238..SR 245 содержат флаги состояния связи систем SYSMAC LINK /SYSMAC NET. Структура данных зависти от системы, используемой для создания связи DATA LINK.

**SYSMAC LINK**

Уровень работы 0	Уровень работы 1	Бит			
		12..15	11..08	04..07	00..03
SR 238	SR 242	Узел 4	Узел 3	Узел 2	Узел 1
SR 239	SR 243	Узел 8	Узел 7	Узел 6	Узел 5
SR 244	SR 244	Узел 12	Узел 11	Узел 10	Узел 9
SR 245	SR 245	Узел 16	Узел 15	Узел 14	Узел 13

Самый левый бит		Самый правый бит	
1: Работа DATA LINK	1: Ошибка связи	1: Ошибка ЦПУ ПК	1: Состояние ПК RUN

**SYSMAC NET**

Уровень работы 0					
		SR 238	SR 239	SR 240	SR 241
Уровень работы 1					
	Бит	SR 242	SR 243	SR 244	SR 245
1: Ошибка ЦПУ ПК	15	8	16	24	32
	14	7	15	23	31
	13	6	14	22	30
	12	5	13	21	29
	11	4	12	20	28
	10	3	11	19	27
	09	2	10	18	26
1: Состояние ПК RUN	08	1	9	17	25
	07	8	16	24	32
	06	7	15	23	31
	05	6	14	22	30
	04	5	13	21	29
	03	4	12	20	28
	02	3	11	19	27
01	2	10	18	26	

Уровень работы 0					
	00	1	9	17	25

### 3.4.2 Системы удаленных входов/выходов

SR 25312 включается в 1 для индикации ошибки систем удаленных входов/выходов. Индикатор ALM/ERR будет мигать, но работа ПК продолжится. AR 251 содержит информацию об источнике и типе ошибки, а AR 0014 и AR 0015 содержат информацию о состоянии SYSMAC LINK. Функция каждого бита описана ниже. Подробности см. Инструкцию по оптическим и проводным удаленным системам входов/выходов.

#### SR 25100 - бит проверки ошибок

Если есть ошибки в нескольких модулях удаленных входов/выходов, в SR 251 будет содержаться информация только о первой ошибке. Данные об остальных модулях будут сохранены в памяти и к ним можно получить доступ включая в 1 и 0 бит проверки ошибки. Обязательно запишите показания первой ошибки, которые сотрутся для индикации следующей.

#### SR 25101 и SR25102

Не используются

#### SR 25103

Флаг ошибки удаленных входов/выходов: бит 03 устанавливается в 1 при появлении признака ошибки в модуле удаленных входов/выходов.

#### SR25104..SR25115

Содержимое битов 04..06 - это трехразрядное двоичное число (04: 20, 05:21, 06: 22), а содержимое битов 08..015 - двухразрядное 16-ричное число (08..11: 160, 12..15: 161).

Если содержимое битов 12..15 = В, произошла ошибка в ведущем или ведомом модуле, и содержимое битов 08..11 будет указывать номер ведущего модулей, 0 или 1. В таком случае биты 04..06 содержат номер ведомой панели.

Если содержимое битов 12..15 = число от 0 до 31, ошибка произошла в модуле оптических входов/выходов или терминале входов/выходов. Это число является номером модулей оптических входов/выходов или терминала входов/выходов, а бит 04 будет в 1, когда модулю выделены левые биты ( 08..15) и 0, если ему выделены правые биты (00..07).

### 3.4.3 Флаги и биты управления системы связи.

Использование следующих битов SR зависит от конфигурации систем связи, в которую встроен Ваш ПК. Эти Флаги и биты управления используются, когда модули связи, такие, как модули PC LINK, модули удаленных входов/выходов или модули HOST LINK смонтированы на панелях ПК или ЦПУ. Дополнительную информацию см. инструкцию по системе для конкретного модулей.

Следующие биты можно использовать в качестве рабочих бит, когда ПК не входит в систему связи, связанную с ними.

#### Системы HOST LINK

Для системы HOST LINK имеются флаги ошибок, и биты перезапуска. Флаги ошибок включаются в 1 для указания наличия ошибки на модулях HOST LINK. Биты перезапуска включают в 1 и затем в 0 для перезапуска Модулей HOST LINK. Биты SR, используемые в системах HOST LINK, обобщены в следующей таблице. Биты перезапуска Модулей HOST LINK, смонтированного на панели, не действуют для многоуровневых Модулей HOST LINK, смонтированного на панели. Подробности см. Инструкцию по системе HOST LINK.

Бит	Флаг
25206	Флаг ошибки Модулей связи HOST LINK, установленного на панели (уровень 1)

Бит	Флаг
25207	Бит перезапуска Модулей связи HOST LINK, установленного на панели (уровень 1)
25213	Бит перезапуска Модулей связи HOST LINK, установленного на панели (уровень 0)
25311	Флаг ошибки Модулей связи HOST LINK, установленного на панели (уровень 0)

### Системы PC LINK

#### Флаги ошибок и работы Модулей PC LINK

Когда ПК встроен в систему связи PC LINK, слова 247..250 используются для просмотра рабочего состояния всех модулей PC LINK, подключенных к системе PC LINK. Система включает максимум 32 модулей PC LINK. Если ПК включен в многоуровневую систему связи PC LINK, половина модулей PC LINK будет в подсистеме PC LINK уровня 0, другая половина в подсистеме PC LINK уровня 1. Фактическое распределение битов зависит от того, находится ПК на одноуровневой или многоуровневой системе связи PC LINK. Подробности см. Инструкцию по системе PC LINK. Присвоения битов флагам ошибок и RUN показаны далее в таблице.

Биты 00..07 каждого слова являются флагами работы (RUN), которые находятся в состоянии 1, когда Модуль PC LINK находится в режиме RUN. Биты 08..15 каждого слова являются флагами ошибки, которые находятся в состоянии 1, когда в модуле PC LINK появляется признак ошибки. В следующей таблице показано присвоение для одноуровневой и многоуровневой систем PC LINK.

#### Одноразовая система PC-Link

Тип флага	Номер бита	SR 247	SR 248	SR 249	SR 250
Флаги работы (RUN)	00	Модуль #24	Модуль #16	Модуль #8	Модуль #0
	01	Модуль #25	Модуль #17	Модуль #9	Модуль #1
	02	Модуль #26	Модуль #18	Модуль #10	Модуль #2
	03	Модуль #27	Модуль #19	Модуль #11	Модуль #3
	04	Модуль #28	Модуль #20	Модуль #12	Модуль #4
	05	Модуль #29	Модуль #21	Модуль #13	Модуль #5
	06	Модуль #30	Модуль #22	Модуль #14	Модуль #6
	07	Модуль #31	Модуль #23	Модуль #15	Модуль #7
Флаги ошибки	00	Модуль #24	Модуль #16	Модуль #8	Модуль #0
	01	Модуль #25	Модуль #17	Модуль #9	Модуль #1
	02	Модуль #26	Модуль #18	Модуль #10	Модуль #2
	03	Модуль #27	Модуль #19	Модуль #11	Модуль #3
	04	Модуль #28	Модуль #20	Модуль #12	Модуль #4
	05	Модуль #29	Модуль #21	Модуль #13	Модуль #5
	06	Модуль #30	Модуль #22	Модуль #14	Модуль #6
	07	Модуль #31	Модуль #23	Модуль #15	Модуль #7

#### Многоуровневая система PC-Link

Тип флага	Номер бита	SR 247	SR 248	SR 249	SR 250
Флаги работы (RUN)	00	Модуль #8, уровень 1	Модуль #0, уровень 1	Модуль #8, уровень 0	Модуль #0, уровень 0
	01	Модуль #9, уровень 1	Модуль #1, уровень 1	Модуль #9, уровень 0	Модуль #1, уровень 0
	02	Модуль #10, уровень 1	Модуль #2, уровень 1	Модуль #10, уровень 0	Модуль #2, уровень 0
	03	Модуль #11, уровень 1	Модуль #3, уровень 1	Модуль #11, уровень 0	Модуль #3, уровень 0



### 3.3 Область IR (внутренние реле)

Тип флага	Номер бита	SR 247	SR 248	SR 249	SR 250
	04	Модуль #12, уровень 1	Модуль #4, уровень 1	Модуль #12, уровень 0	Модуль #4, уровень 0
	05	Модуль #13, уровень 1	Модуль #5, уровень 1	Модуль #13, уровень 0	Модуль #5, уровень 0
	06	Модуль #14, уровень 1	Модуль #6, уровень 1	Модуль #14, уровень 0	Модуль #6, уровень 0
	07	Модуль #15, уровень 1	Модуль #7, уровень 1	Модуль #15, уровень 0	Модуль #7, уровень 0
Флаги ошибки	08	Модуль #8, уровень 1	Модуль #0, уровень 1	Модуль #8, уровень 0	Модуль #0, уровень 0
	09	Модуль #9, уровень 1	Модуль #1, уровень 1	Модуль #9, уровень 0	Модуль #1, уровень 0
	10	Модуль #10, уровень 1	Модуль #2, уровень 1	Модуль #10, уровень 0	Модуль #2, уровень 0
	11	Модуль #11, уровень 1	Модуль #3, уровень 1	Модуль #11, уровень 0	Модуль #3, уровень 0
	12	Модуль #12, уровень 1	Модуль #4, уровень 1	Модуль #12, уровень 0	Модуль #4, уровень 0
	13	Модуль #13, уровень 1	Модуль #5, уровень 1	Модуль #13, уровень 0	Модуль #5, уровень 0
	14	Модуль #14, уровень 1	Модуль #6, уровень 1	Модуль #14, уровень 0	Модуль #6, уровень 0
	15	Модуль #15, уровень 1	Модуль #7, уровень 1	Модуль #15, уровень 0	Модуль #7, уровень 0

#### Пример применения

Если ПК находится в многоуровневой системе связи PC LINK и содержание слова 248 = 02FF, тогда модули PC LINK #0..#7, назначенные на подсистему уровня 1, будут в режиме RUN (работа) и в модуль #1 той же подсистемы будет ошибка. 16-ричные цифры и соответствующие двоичные биты слова 248 будут иметь следующий вид:

Номер бита	15													00		
Двоичный	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1
16-ричный	F	2				F				F						

#### 3.4.4 Бит удержания принудительного состояния

SR 25211 определяет, сохранять ли принудительное состояние бита при переключении между режимами PROGRAM и MONITOR или пуске и останове отработки программы. Если SR 25211 = 1, состояние бита будет сохранено; если SR 25211 = 0, все биты будут возвращены в состояние по умолчанию, когда отработка программы начинается или оканчивается. Бит удержания принудительного состояния эффективен только когда есть разрешение установочных параметров ПК.

На состояние SR 25211 прерывание питания влияет только когда таблица входов/выходов зарегистрирована. В таком случае SR 25211 установится в 0.

SR 25211 не действует при переключении в режим RUN.

Управление битом SR 25211 осуществляется с периферийного устройства, например, программатора или ПО.

#### Сохранение состояния при пуске

Состояние SR 25211 и, следовательно, состояние принудительно установленных и сброшенных битов, можно сохранить при отключении и повторном включении питания путем разрешения в установочных параметрах ПК. Если действие бита сохранения принудительного состояния разрешено, состояние SR 25211 будет сохраняться при отключении и включении питания. Если это произошло и SR 25211 = 1, тогда состояние принудительно установленных и сброшенных битов, будет сохранено, как показано в таблице.

Состояние перед выключением	Состояние при следующем пуске	
SR 25211	SR 25211	Принудительно установленные/ сброшенные биты
1	1	Состояние сохраняется
0	0	Сброс

**Замечание** Подробности о разрешении действия бита сохранения принудительного состояния см. 3.6.4.

### 3.4.5 Бит удержания состояния входов/выходов

SR 25212 определяет, сохранять состояние битов области IR и LR при пуске и останове отработки программы, когда отработка начинается переключением из режима PROGRAM в MONITOR или RUN. Если SR 25212 = 1, состояние бита будет сохранено; если SR 25212 = 0, все биты области IR и LR будут сброшены. Бит удержания состояния входов/выходов эффективен только когда есть разрешение установочных параметров ПК.

На состояние SR 25212 прерывание питания влияет только когда таблица входов/выходов зарегистрирована. В таком случае SR 25212 установится в 0.

Управление битом SR 25212 осуществляется с периферийного устройства, например, программатора или ПО.

#### Сохранение состояния при пуске

Состояние SR 25212 и, следовательно, состояние битов области IR и LR можно сохранить при отключении и повторном включении питания путем разрешения в установочных параметрах ПК. Если действие бита сохранения состояния разрешено, состояние SR 25212 будет сохраняться при отключении и включении питания. Если это произошло и SR 25212 = 1, тогда состояние битов IR и LR будет также сохранено, как показано в таблице.

Состояние перед выключением	Состояние при следующем пуске	
SR 25212	SR 25212	Биты IR и LR
1	1	Состояние сохраняется
0	0	Сброс

**Замечание** Подробности о разрешении действия бита сохранения состояния входов/выходов см. 3.6.4.

### 3.4.6 Бит сброса выходов

Бит SR 25215 включается в 1 для сброса в 0 всех выходов ПК. При этом загорится индикатор на передней панели ЦПУ OUT INHIBIT. Когда бит сброса выходов = 0, все биты выходов будут обновляться обычным путем.

Состояние бита сброса выходов сохраняется на время прерываний питания или при останове отработки программы, если только таблица входов/выходов не зарегистрирована, или таблица входов/выходов зарегистрирована и действие битов (сохранение принудительного состояния или сохранения входов/выходов) разрешены в установочных параметрах ПК.

### 3.4.7 Область FAL (предупреждение об ошибке)

2-разрядный код FAL выдается в биты 25300..25307 после выполнения команд FAL или FALS. Данные коды определяются пользователем для диагностики ошибок., ПК также выдает коды FAL в эти биты, такие как вызванные падением напряжения аккумулятора.

Данную область можно очистить выполнением команды FAL с операндом 00 или выполнив операцию чтения признаков ошибок с программатора.

### 3.4.8 Флаг падения напряжения батареи

SR 25308 включается в 1 при падении напряжения батареи на ЦПУ. Индикатор ARM/ERR на передней панели ЦПУ будет мигать.

Данный бит можно запрограммировать для вызова внешних сигналов предупреждения о падении напряжения батареи.

При желании, операцию предупреждения о падении напряжении батареи можно запретить из установочных параметров ПК. Подробности см 3.6.4.

### 3.4.9 Флаг ошибки времени цикла

SR 25309 включается в 1, если время цикла превысило 100 мс. Индикатор ARM/ERR на передней панели ЦПУ будет мигать. Исполнение программы не прекратится, если только не превышен предел максимального времени контрольного таймера. Синхронизация может быть нарушена при превышении времени цикла 100 мс.

### 3.4.10 Флаг ошибки сверки входов/выходов

Бит SR 25310 включается в 1, когда модули, установленные в системе, не согласуются с таблицей входов/выходов, зарегистрированной в ЦПУ. Индикатор ARM/ERR на передней панели ЦПУ будет мигать, но отработка программы ПК не будет продолжаться.

При включении флага в 1 для обеспечения правильности отработки программы ее следует остановить, проверить модули и скорректировать таблицу входов/выходов.

### 3.4.11 Флаг первого цикла

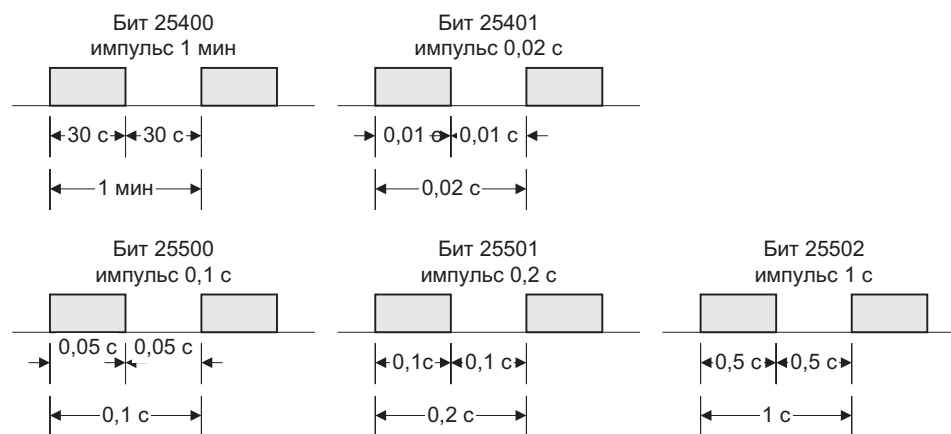
SR 25315 включается в 1, при пуске отработки программы ПК и выключается в 0 после выполнения одного цикла программы. Флаг первого цикла полезен для инициализации значений счетчиков и других операций. Пример показан в 5.14.

### 3.4.12 Биты синхронизирующих импульсов

Для управления синхронизацией программы предусмотрены 5 бит синхронизирующих импульсов. Бит каждого синхронизирующего импульса = 1 в первую половину времени импульса, затем = 0 во вторую половину. Другими словами, каждый импульс имеет коэффициент заполнения 50 %.

Эти биты синхронизирующих импульсов часто используются в комбинации со счетчиками для создания таймеров. Примеры см. 5.14.

Ширина импульса	1 мин	0.02 с	0.1 с	0.2 с	1.0 с
Бит	25400	25401	25500	25501	25502



**Замечание** Ввиду того, что импульсы длительностью 0,1с и 0,02с находятся в состоянии 1 (ON) 50 и 10мс соответственно, ЦПУ может не совсем Модуль прекращает работу и мигает индикатор ALARM, но ПК продолжает отработку программы.

Когда флаг ошибки группы 2 = 1, номер модулей с ошибкой будет находиться в AR 0205..AR 0214. Если Модуль нельзя запустить должным образом, хотя номер входов/выходов задан корректно и Модуль смонтирован правильно, может быть, сгорел предохранитель или произошла ошибка в аппаратной части. Если это произошло, замените Модуль на запасной и попытайтесь снова запустить систему.

Имеется также флаг ошибки модулей входов/выходов высокой плотности и модулей интерфейса B7A в области AR, а именно AR 0215.

#### 3.4.15 Флаг ошибки специальных модулей

SR 25415 включается в 1 для индикации ошибок в следующих модулях: специальных входов/выходов, PC LINK, HOST LINK и ведущих удаленных входов/выходов. SR 25415 включается в 1 при одной из следующих ошибок:

- Когда на один номер задано несколько модулей специальных входов/выходов.
- Когда ошибка происходит при обновлении данных между модулем специальных входов/выходов и ЦПУ ПК.
- Когда ошибка происходит между модулем HOST LINK и ЦПУ ПК.
- Когда ошибка происходит в модуле ведущих удаленных входов/выходов.

Хотя ПК продолжает отработку программы, если SR 25415 =ON, модули, вызвавшие ошибку, прекратят работу и будет мигать индикатор ALM. Проверьте состояние AR 000..AR 0015, чтобы узнать номер модулей с ошибкой и найдите причину ошибки.

Работу модулей можно возобновить битами перезапуска (AR 0100..AR 0115, SR 25207..SR 25213), но это не поможет, если одинаковый номер модулей задан для более чем одного модулей специальных входов/выходов. Выключите питание, исправьте задание модулей и снова включите питание.

SR 25415 не выключится в 0 даже если AR 0100..AR 0115 (биты перезапуска) включатся в ON. Его можно выключить в 0 путем чтения ошибок с программирующих устройств или исполнением команды FAL 00 из релейно-контактной программы.

#### 3.4.16 Флаг ошибки исполнения команды, ER

SR 25503 включается в 1, если была сделана попытка выполнить команду с некорректным операндом. Общие причины ошибки команды - это операнд не в двоично-десятичном виде, когда это требуется, или косвенный адрес DM не существует.

**Внимание!** Когда флаг ER = 1, текущая команда не выполняется.

#### 3.4.17 Флаги арифметики

Следующие флаги используются в командах сдвига, математических вычислений и сравнения. К ним обычно обращаются по их двум первым буквам.

**Замечание** Эти флаги все сброшены, когда выполняется команда END и, следовательно, их нельзя просмотреть с программирующего устройства.

Подробности см. 5.15, 5.19, и 5.20

##### Флаг отрицательного значения (N)

SR 25402 включается в 1, когда результат вычисления отрицателен.

##### Флаг переполнения + (OF)

SR 25404 включается в 1, когда результат двоичного сложения или вычитания превысит 7FFF или 7FFF FFFF.

##### Флаг переполнения - (UF)

SR 25405 включается в 1, когда результат сложения или вычитания двоичных чисел со знаком превысит 8000 или 8000 0000.

##### Флаг переноса (CY)

SR 25504 включается в 1, когда имеется перенос в результате математической операции или когда команды сдвига помещают 1 в CY. Содержимое CY также используется в

некоторых математических операциях, например, оно складывается или вычитается с другими операндами. Этот флаг можно задать и очистить из программы командами SET CARRY и CLEAR CARRY.

#### Флаг больше чем (GR)

SR 25505 включается в 1, когда результат сравнения показывает, что первый из двух операндов больше второго.

#### Флаг равно (EQ)

SR 25406 включается в 1, когда результат сравнения показывает, что операнды равны или результат арифметической операции равен нулю.

#### Флаг меньше чем (LE)

SR 25507 включается в 1, когда результат сравнения показывает, что первый из двух операндов меньше второго.

**Замечание** Эти 4 флага арифметики включаются в 0 после выполнении команды END.

### 3.4.18 Области подпрограмм прерывания

Следующие области используются при обработке подпрограмм прерывания.

#### Область максимального времени обработки подпрограмм прерывания

SR 26200..SR 26215 используются для задания максимального времени обработки подпрограмм прерывания. Время обработки задается в 0.1 секунд.

#### Область номера подпрограммы прерывания с максимальным временем обработки

SR 26300..SR 26315 содержит номер подпрограмм прерывания с максимальным временем обработки. Бит 15 будет = 1 при прерывании.

### 3.4.19 Области связи порта RS-232C

#### Код ошибки порта RS-232C

SR 26400..SR 26403 устанавливаются в 1 при ошибке порта RS-232C.

Задание	Тип ошибки
0	Нет ошибки
1	Ошибка четности
2	Ошибка формата
3	Ошибка переполнения
4	Ошибка КСК
5	Ошибка времени ожидания
6	Ошибка контрольной суммы
7	Ошибка команды

#### Бит ошибки связи порта RS-232C

SR 26404 устанавливаются в 1 при ошибке связи порта RS-232C.

#### Флаг готовности передачи порта RS-232C

SR 26405 устанавливаются в 1, когда порт RS-232C готов к передаче данных.

#### Флаг завершения приема с порта RS-232C

SR 26406 устанавливается в 1, когда ПК завершил чтение с устройства, подключенного к порту RS-232C.

#### Флаг переполнения приема с порта RS-232C

SR 26407 устанавливается в 1 при переполнении при приеме данных.

#### Счетчик приема порта RS-232C

SR 26500..SR 26515 содержат число приемов порта RS-232C в общем режиме входов/выходов.

#### Флаг готовности передачи HOST LINK уровня 0

SR 26705 устанавливается в 1, когда ПК готов к передаче на Модуль HOST LINK.

**Флаг готовности передачи HOST LINK уровня 1**

SR 26713 устанавливается в 1, когда ПК готов к передаче на HOST LINK.

**3.4.20 Области связи периферийного порта****Код ошибки периферийного порта**

SR 26408..SR 26411 устанавливаются в 1 при ошибке периферийного порта в режиме общих входов/выходов.

Задание	Тип ошибки
0	Нет ошибки
1	Ошибка четности
2	Ошибка формата
3	Ошибка переполнения
4	Ошибка КСК
5	Ошибка времени ожидания
6	Ошибка контрольной суммы
7	Ошибка команды

**Бит ошибки связи периферийного порта**

SR 26412 устанавливаются в 1 при ошибке связи периферийного порта (действует в режиме общего ввода/вывода).

**Флаг готовности передачи периферийного порта**

SR 26413 устанавливаются в 1, когда периферийный порт готов к передаче данных (действует в режиме общего ввода/вывода)..

**Флаг завершения приема с периферийного порта**

SR 26414 устанавливается в 1, когда ПК завершил чтение с устройства на периферийном порту. (действует в режиме общего ввода/вывода)..

**Флаг переполнения приема с периферийного порта**

SR 26415 устанавливается в 1 при переполнении при приеме данных. (Действует в режиме общего ввода/вывода)..

**Счетчик приема порта RS-232C**

SR 26600..SR 26615 содержат число приемов данных периферийного порта в режиме общего ввода/вывода.

**Флаг готовности передачи HOST LINK уровня 0**

SR 26705 устанавливается в 1, когда ПК готов к передаче на Модуль HOST LINK.

**Флаг готовности передачи HOST LINK уровня 1**

SR 26713 устанавливается в 1, когда ПК готов к передаче на HOST LINK.

**3.4.21 Области кассеты памяти****Содержимое кассеты памяти**

SR 26900..SR 26907 указывает на тип памяти, используемый для кассеты памяти.

Тип памяти	Код
Нет	00
UM	01
ИОМ	02
HIS	03

**Емкость кассеты памяти**

SR 26908..SR 26910 указывает емкость кассеты памяти.

Емкость, К слов	Код
0	0

Емкость, К слов	Код
4/8	2
16	3
32	4

**Флаг кассеты EEPROM/EPROM**

SR 26914 включается в 1, когда кассета памяти EEPROM защищена от записи или установлена кассета памяти с EPROM.

**Флаг наличия кассеты**

SR 26915 включается в 1, когда кассета памяти установлена.

**Флаг сохранения UM на кассету**

SR 27000 включается в 1, когда данные UM читаются в кассету памяти в программном режиме. Бит автоматически сбросится в 0. Если включить в 1 в любом другом режиме, появится признак ошибки.

**Флаг загрузки UM с кассеты**

SR 27001 включается в 1, когда данные загружаются в UM из кассеты памяти в программном режиме. Бит автоматически сбросится в 0. Если включить в 1 в любом другом режиме, появится признак ошибки.

**Сравнение (DM и кассетой памяти)**

SR 27002 включается в 1, когда сравниваются данные между DM и кассетой памяти. SR 27003 включается в 0, если данные в кассете памяти совпадают, и 1, если не совпадают.

**3.4.22 Биты ошибки передачи**

Данные не будут переданы из UM в кассету памяти в случае ошибки (за исключением ошибки контрольной суммы панели). Подробная информация об ошибках контрольной суммы, которые случаются в кассете памяти, не будут посланы в SR 272 из-за того, что она не нужна. Повторите передачу, если SR 27015 = 1.

**Флаг ошибки передачи: активна связь DATA LINK**

SR 27011 включается в 1, при попытке передачи UM, используемой для таблицы данных SYSMAC NET, когда активна DATA LINK.

**Флаг ошибки передачи: не режим PROGRAM**

SR 27012 включается в 1 при попытке передачи данных, когда ПК не в режиме PROGRAM.

**Флаг ошибки передачи: только чтение**

SR 27013 включается в 1 при попытке передачи данных, когда ПК работает в режиме только чтения.

**Флаг ошибки передачи: Недостаточно памяти или не UM**

SR 27014 включается в 1 при попытке передачи данных, когда недостаточен объем UM.

**Флаг ошибки передачи: Ошибка контрольной суммы панели.**

SR 27015 включается в 1 при попытке передачи данных, когда произошла ошибка контрольной суммы панели.

**3.4.23 Области памяти программы в виде РКС**

**Область размера РКС на кассете памяти**

SR 27100..SR 27107 указывают на количество программ (в виде РКС) на кассете памяти. Только файлы РКС:

04: 4 К слов, 08: 8 К слов, 12: 12 К слов..(32: 32 К слов)

00: На кассете памяти не UM, или файла нет.

**Размер и тип РКС на ЦУ**

SR 27108..SR 27115 указывают на тип и размер РКС в ЦУ. Значения те же, что и для битов 00..07.

#### 3.4.24 Флаги ошибки памяти

**Флаг ошибки памяти: Ошибка установочных параметров ПК**

SR 27211 включается в 1 при ошибке контрольной суммы установочных параметров.

**Флаг ошибки памяти: Ошибка контрольной суммы релейно-контактной программы**

SR 27212 включается в 1 при ошибке контрольной суммы релейно-контактной программы.

**Флаг ошибки памяти: Ошибка изменения команды**

SR 27213 включается в 1 при ошибке области вектора изменения команды.

**Флаг ошибки памяти: ошибка отключения кассеты**

SR 27214 включается в 1, когда кассета памяти отключается или подключается во время операции.

**Флаг ошибки памяти: ошибка автозагрузки**

SR 27215 включается в 1 при ошибке автозагрузки.

#### 3.4.25 Флаги сохранения данных

Данные переданы на кассету памяти, когда бит включается в 1 в режиме PROGRAM. Бит автоматически сбрасывается в 0. Ошибка появится при включении в 1 в любом другом режиме.

**Бит сохранения IOM на кассету**

SR 27300 включается в 1, когда IOM сохраняется на кассете памяти.

**Бит загрузки IOM с кассеты**

SR 27301 включается в 1, когда IOM загружается с кассеты памяти.

#### 3.4.26 Флаги ошибки передачи

Данные не будут переданы с IOM на кассету памяти при ошибке (за исключением ошибки только чтение).

**Флаг ошибки передачи: Режим не PROGRAM**

SR 27312 включается в 1 при попытке передачи данных в режиме не PROGRAM.

**Флаг ошибки передачи**

SR 27313 включается в 1 при попытке передачи данных в режиме ТОЛЬКО ЧТЕНИЕ.

**Флаг ошибки передачи**

SR 27314 включается в 1 при попытке передачи данных, когда недостаточно памяти IOM.

#### 3.4.27 Флаги ошибки установочных параметров ПК

**Флаги ошибки установочных параметров ПК пуска**

SR 27500 включается в 1 при ошибке установочных параметров пуска (DM 6600..DM 6605).

**Флаги ошибки установочных параметров ПК при работе (состояние RUN)**

SR 27501 включается в 1 при ошибке установочных параметров при работе (DM 6613..DM 6623).

**Ошибка параметров связи, параметров задания ошибок и остальные**

SR 27501 включается в 1 при ошибке параметров связи, параметров задания ошибок и остальных ошибках (DM 6635..DM 6655).

#### 3.4.28 Текущее время и отображение состояния клавиатуры

**Часы (SR 276)**

SR 276 содержит текущее время. SR 27600..SR 27607 содержит минуты (00..59) и SR 27608..SR 27615 содержит часы (0..23).



#### Отображение состояния клавиатуры (SR 277)

SR 277..SR 279 используются для отображение состояния клавиатуры.

#### 3.4.29 Флаги ошибки группы 2

SR 28000..SR 28015 используются в качестве флагов ошибок модулей входов/выходов высокой плотности группы 2 с номерами модулей 0..F. Соответствующий флаг ошибки включается в 1 при ошибке в этом модуле. 10 бит области AR (AR 0205..AR 0214) также используются в качестве флагов ошибок для модулей 0..9.

#### 3.4.30 Биты перезапуска и флаги ошибок Специальных модулей

Биты SR 28100..SR 28115 используются в качестве битов перезапуска для модулей специальных входов/выходов с номерами 0..F. Включите соответствующий бит в 1 и 0 для пуска модулей специальных входов/выходов. 10 бит области AR (AR 0100..AR 0109) также используются в качестве битов перезапуска для модулей 0..9.

Биты SR 28200..SR 28215 используются в качестве флагов ошибок для модулей специальных входов/выходов с номерами 0..F. Соответствующий флаг ошибки включается в 1 при ошибке в этом модуле. 10 битов области AR (AR 0000..AR 0009) также используются в качестве флагов ошибок для модулей 0..9.

### 3.5 Область AR (Вспомогательных реле)

Адреса слов AR располагаются в зоне AR 00..AR 27; Адреса битов AR располагаются в зоне AR 0000..AR 2715. Большинство слов и битов предназначены для особых задач, таких как счетчики передачи, флаги и биты управления, а слова AR 00..AR 07 и AR 23..AR 27 нельзя использовать для других целей. Слова и биты AR 08..AR 15 можно использовать в качестве рабочих слов и битов, если они не используются для следующих целей.

Слово	Использование
AR 08..AR 15	модули SYSMAC LINK
AR 16, AR 17	модули связи SYSMAC LINK и SYSMAC NET

Область AR сохраняет состояние при прерывании питания, при переключении из режимов MONITOR или RUN в PROGRAM, или при останове отработки программы. Распределение битов показано в следующей таблице в порядке номеров битов.

Флаги области AR и биты управления

Слово (слова)	Бит(ы)	Функция
00	00..09	Флаги ошибок для модулей специальных входов/выходов 0..9 и модулей PC LINK 0..9 (Функция этих флагов дублированы в SR 28200..SR 28209)
	10	Флаг ошибки для рабочего уровня 1 системы SYSMAC LINK или SYSMAC NET
	11	Флаг ошибки для рабочего уровня 0 системы SYSMAC LINK или SYSMAC NET
	12	Флаг ошибки передачи с управляющего компьютера в Модуль HOST LINK, установленный на панели (для рабочего уровня 1)
	13	Флаг ошибки передачи с управляющего компьютера в Модуль HOST LINK, установленный на панели (для рабочего уровня 0)
	14	Флаг ошибки 1 ведущего модулей удаленных входов/выходов
	15	Флаг ошибки 0 ведущего модулей удаленных входов/выходов
01	00..09	Биты перезапуска для модулей специальных входов/выходов 0..9 и модулей PC LINK 0..9 (Работа этих битов дублирована в SR 28100..28109)
	10	Бит перезапуска рабочего уровня 1 системы связи SYSMAC LINK или SYSMAC NET

### 3.5 Область AR (Вспомогательных реле)

Слово (слова)	Бит(ы)	Функция
	11	Бит перезапуска рабочего уровня 0 системы связи SYSMAC LINK или SYSMAC NET
	12..13	Не используются
	14	Бит перезапуска ведущего модулей 1 удаленных входов/выходов
	15	Бит перезапуска ведущего модулей 0 удаленных входов/выходов
02	00..04	Флаги ошибки ведомой панели (#0..#4)
	05..14	Флаги ошибки группы 2 (биты 05..14 соответствуют модулям 0..9)
	15	Флаги ошибки группы 2
03	00..15	Флаги ошибок для модулей оптических входов/выходов и терминалов входов/выходов 0..7
04	00..15	Флаги ошибок для модулей оптических входов/выходов и терминалов входов/выходов 8..15
05	00..15	Флаги ошибок для модулей оптических входов/выходов и терминалов входов/выходов 16..23
06	00..15	Флаги ошибок для модулей оптических входов/выходов и терминалов входов/выходов 24..31
07	00..03	Установочные параметры рабочего уровня 0 системы SYSTEM LINK
	04..07	Установочные параметры рабочего уровня 1 системы SYSTEM LINK
	08	Бит отмены ввода режима TERMINAL
	09..11	Не используются
	12	Аналогично состоянию секции 6 переключателя DIP ЦПУ
	13	Бит перезаписи протокола ошибок
	14	Бит сброса протокола ошибок
15	Бит разрешения протокола ошибок	
08..11	00..15	Флаги активного узла для системы SYSMAC LINK рабочего уровня 0
12..15	00..15	Флаги активного узла для системы SYSMAC LINK рабочего уровня 1
16	00..15	Система SYSMAC LINK/SYSMAC NET, уровень 0, время обслуживания на цикл
17	00..15	Система SYSMAC LINK/SYSMAC NET, уровень 1, время обслуживания на цикл
18	00..07	Секунды: 00..59
	08..15	Минуты: 00.. 59
19	00..07	Часы: 00..24 (система 24 часа)
	08..15	День месяца: 01..31 (настроен на месяц и на високосный год)
20	00..07	Месяц: 1..12
	08..15	Год: 00..99 (две младшие цифры года)
21	00..07	День недели: 00..06 (00 - вс, 01 - пн, 02 - вт, 03 - ср, 04 - чт, 05- пт, 06 - сб)
	08..12	Не используется
	13	Бит компенсации 30 с
	14	Бит останова часов
	15	Бит установки часов
22	00..15	Отображение клавиатуры
23	00..15	Счетчик отключений питания (двоично-десятичный)
24	00	SYSMAC LINK - Флаг периферийного устройства на RS-232C
	01	SYSMAC LINK - Флаг периферийного устройства порта А
	02	SYSMAC LINK - Флаг периферийного устройства порта В
	03	SYSMAC LINK - Бит инициализации периферийного устройства
	04	Не используется

Слово (слова)	Бит(ы)	Функция
	05	Флаг времени цикла
	06	Флаг параметра сети SYSMAC LINK (рабочего уровня 1)
	07	Флаг параметра сети SYSMAC LINK (рабочего уровня 0)
	08	Флаг "Модуль связи SYSMAC LINK/ SYSMAC NET рабочего уровня 1 смонтирован"
	09	Флаг "Модуль связи SYSMAC LINK/ SYSMAC NET рабочего уровня 0 смонтирован"
	10..12	Не используется
	13	Флаг "Модуль HOST LINK рабочего уровня 1, монтаж на панели, установлен"
	14	Флаг "Модуль HOST LINK рабочего уровня 0, монтаж на панели, установлен"
	15	Флаг "устройство, монтаж на ЦПУ, установлено"
25	00..07	Бит запрета пароля доступа редактирования в режиме ON-LINE (Бит запрета редактирования в режиме ON-LINE действует при значении байта 5A)
	08	Бит переключения FPD
	09	Бит запрета редактирования в режиме ON-LINE
	10	Флаг резервного редактирования в режиме ON-LINE
	11	Не используется
	12	Флаг Конец трассировки
	13	Флаг трассировки
	14	Бит переключения трассировки (можно писать)
	15	Бит начала трассировки (можно писать)
26	00..15	Время максимального цикла (0.1 мс)
27	00..15	Время текущего цикла (0.1 мс)

#### 3.5.1 Перезапуск модулей специальных входов/выходов

Биты AR 0100..AR 109 соответствуют номерам модулей специальных входов/выходов 0..9. Для перезапуска модулей специальных входов/выходов (включая модули PC LINK) включите соответствующие модули в 1 и 0 (или включите и выключите питание). Не обращайтесь к обновленным данным модулей специальных входов/выходов во время процесса перезапуска (см. 3.4, SR 27400..SR 27409).

**Замечание** Биты SR 28100..SR 28115 также действуют как биты перезапуска для модулей специальных входов/выходов 0..F.

#### 3.5.2 Флаги ошибки ведомой панели

Биты AR 0200..AR 0204 соответствуют номерам модулей удаленных ведомых входов/выходов #0..#4. Эти флаги включатся в 1, если одинаковый номер выделен больше, чем одному ведомому или при ошибке передачи при пуске системы. Об ошибках, появляющихся, когда система стартует нормально, см. SR 251.

#### 3.5.3 Флаги ошибки группы 2

Биты AR 0205..AR 0215 соответствуют модулям входов/выходов высокой плотности группы 2 и модулям 0..9 интерфейса B7A (номера входов/выходов) и будут включены в 1, когда одинаковый номер задан более одному модулю или одно слово будет выделено более одному модулю, когда номер входов/выходов 9 установлен на Модуль 64 точки, или когда перегорел предохранитель в транзисторном модуле входов/выходов высокой плотности. AR 0215 включится в 1, когда Модуль не распознан как Модуль входов/выходов высокой плотности группы 2.

**Замечание** Биты SR 28000..SR 28015 также действуют как флаги ошибки модулей входов/выходов высокой плотности группы 2 с номерами 0..F.

#### 3.5.4 Флаги ошибок модулей оптических входов/выходов и терминалов входов/выходов

AR 03..AR 06 содержат флаги ошибок модулей оптических входов/выходов и терминалов входов/выходов. Ошибка указывает на дублирование номера модулей. К ПК можно подключить до 64 модулей оптических входов/выходов и терминалов входов/выходов. Модули отличаются по номерам модулей 0..31, и буквам L или H. Выделение битов показано в таблице.

##### Флаги ошибок модулей оптических входов/выходов

Биты	AR 03	AR 04	AR 05	AR 06
00	0 L	8 L	16 L	24 L
01	0 H	8 H	16 H	24 H
02	1 L	9 L	17 L	25 L
03	1 H	9 H	17 H	25 H
04	2 L	10 L	18 L	26 L
05	2 H	10 H	18 H	26 H
06	3 L	11 L	19 L	27 L
07	3 H	11 H	19 H	27 H
08	4 L	12 L	20 L	28 L
09	4 H	12 H	20 H	28 H
10	5 L	13 L	21 L	29 L
11	5 H	13 H	21 H	29 H
12	6 L	14 L	22 L	30 L
13	6 H	14 H	22 H	30 H
14	7 L	15 L	23 L	31 L
15	7 H	15 H	23 H	31 H

#### 3.5.5 Установочные параметры системы связи SYSTEM LINK

AR 0700..AR 0703 и AR 0704.. AR 0707 используются для присвоения словам рабочего уровня 0 и 1 системы SYSTEM LINK. Присвоение можно задать либо согласно заданию с ПО, либо автоматически в областях LR и/или DM. Если задано автоматическое присвоение, число слов, выделенных каждому узлу, также указывается. Эти задания показаны в таблице.

##### Внешнее/ автоматическое распределение

Рабочий уровень 0		Рабочий уровень 1		Значения	
AR 0700	AR 0701	AR 0704	AR 0705		
0	0	0	0	Слова задаются извне (ПО)	
1	0	1	0	Автоматическое распределение	Только область LR
0	1	0	1		Только область DM
1	1	1	1		Области LR и DM

##### Слов на узел

Следующие задания необходимы при автоматическом распределении, описанном ранее.

Уровень 0		Уровень 1		Слов на узел		Макс. число узлов
AR 0702	AR 0703	AR 0706	AR 0707	Область LR	Область DM	
0	0	0	0	4	8	16
1	0	1	0	8	16	8

### 3.5 Область AR (Вспомогательных реле)

Уровень 0		Уровень 1		Слов на узел		Макс. число узлов
0	1	0	1	16	32	4
1	1	1	1	32	64	2

Указанные значения читаются каждый цикл при работе системы SYSMAC LINK.

#### 3.5.6 Биты протокола ошибок

AR 0713 (бит переписи протокола ошибок) включается в 1 или 0 пользователем для управления переписью протокола в области DM. Включите AR 0713 в 1 для переписи самой старой записи при каждой новой ошибке после того, как будет записано 10 ошибок. Включите AR 0713 в 0 для записи только первых 10 записей после очистки области протокола.

AR 0714 (бит сброса протокола ошибок) включается в 1, а затем в 0 пользователем для сброса указателя записи в протоколе (DM 6000) и, таким образом, возобновления записи с начала области протокола.

AR 0715 (бит разрешения протокола ошибок) включается в 1 пользователем для разрешения протокола ошибок и в 0 для запрещения протокола.

Подробности об области протокола ошибок см. 3.6.

Биты протокола ошибок обновляются каждый цикл.

#### 3.5.7 Флаги активного узла

AR 08..AR 11 и AR 12..AR 15 содержат флаги, которые указывают, какие узлы активны в системе SYSMAC LINK в настоящее время. Эти флаги обновляются каждый цикл, когда работает SYSMAC LINK.

В таблице представлен номер узла, присвоенного каждому биту. Если бит = 1, этот узел активен (в таблице указаны номера узлов).

Уровень работы 0				
	AR 08	AR 09	AR 10	AR 11
Уровень работы 1				
Бит	AR 12	AR 13	AR 14	AR 15
00	1	17	33	49
01	2	18	34	50
02	3	19	35	51
03	4	20	36	52
04	5	21	37	53
05	6	22	38	54
06	7	23	39	55
07	8	24	40	56
08	9	25	41	57
09	10	26	42	58
10	11	27	43	59
11	12	28	44	60
12	13	29	45	61
13	14	30	46	62
14	15	31	47	63
15	16	32	48	64

\* - Флаг ошибки контроллера связи

\*\* - Флаг ошибки EEPROM

### 3.5.8 Время обслуживания системы SYSMAC LINK/SYSMAC NET

AR 16 содержит информацию о времени обслуживания уровня 0 системы SYSMAC LINK и/или SYSMAC NET в течение каждого цикла, когда Модуль системы SYSMAC LINK и/или SYSMAC NET смонтирован на панели.

AR 17 содержит информацию о времени обслуживания уровня 1 системы SYSMAC LINK и/или SYSMAC NET в течение каждого цикла, когда Модуль системы SYSMAC LINK и/или SYSMAC NET смонтирован на панели.

Эти времена записываются как 4-разрядное двоично-десятичное число в десятых долях миллисекунды (000.0 мс..999.9 мс) и обновляются каждый цикл.

Биты			
15..12	11..08	07..04	03..00
10 <sup>2</sup>	10 <sup>1</sup>	10 <sup>0</sup>	10 <sup>-1</sup>

### 3.5.9 Области и Биты календарь/часы

#### Область календарь/часы

В ЦПУ С200НХ/НГ/НЕ встроены часы. Если AR 2114 (бит останова часов) = 0, тогда есть доступ к двоично-десятичным значениям времени в AR 18..AR 20 и AR 2100..AR 2108, как показано в таблице. Данная область может управляться битами AR 2113 (бит компенсации 30-секунд) и AR 2115 (бит установки часов).

Биты календарь/часы

Биты	Содержимое	Возможные значения
AR 1800..AR 1807	Секунды	00..59
AR 1808..AR 1815	Минуты	00.. 59
AR 1900..AR 1907	Часы	00..23 (система 24 часа)
AR 1908..AR 1915	День месяца	01..31 (настроен на день недели и на високосный год)
AR 2000..AR 2007	Месяц	1..12
AR 2008..AR 2015	Год	00..99 (младшие цифры года)
AR 2100..AR 2107	День недели	00..06 (00 - вс, 01 - пн, 02 - вт, 03 - ср, 04 - чт, 05 - пт, 06 - сб)

#### Бит компенсации 30-секунд

AR 2113 включается в 1 для округления секунд области календарь/часы. Если 29 секунд и менее, они сбрасываются в 0. Если 30 секунд и более, они сбрасываются в 0, а минуты инкрментуются..

#### Бит останова часов

AR 2114 включается в 0 для разрешения операций области календарь/часы и в 0 для останова.

#### Бит установки часов

AR 2115 служит для задания времени. Эти данные должны быть в двоично-десятичном виде и должны задаваться в допустимых зонах, указанных ранее.

- 1, 2, 3,... 1. Установите AR 2114 в 1 (бит остановки часов) для остановки часов.
2. Установите требуемые дату, день и время, и при установке дня недели случайно не включите в 0 AR 2114 (бит останова) (они находятся в одном слове). На программаторе это легче всего сделать с помощью операций просмотра бита/цифры и принудительной установки/сброса.
3. Установите AR 2115 в 1 (бит установки часов). Часы начнут идти с заданного времени, и биты AR 2114 и AR 2115 сбросятся в 0.

### 3.5.10 Биты клавиш режима TERMINAL

Если Консоль программирования подключен к ПК и находится в режиме TERMINAL, все входы с клавиш 0..9 (включая символы A..F, т.е. клавиши 0..5 через SHIFT) будут

включать соответствующие биты в AR 22. В режим TERMINAL переход осуществляется с помощью операции ПК.

Биты AR 22 соответствуют клавишам программатора следующим образом:

Бит	Ввод с программатора
AR 2200	0
AR 2201	1
AR 2202	2
AR 2203	3
AR 2204	4
AR 2205	5
AR 2206	6
AR 2207	7
AR 2208	8
AR 2209	9
AR 2210	A
AR 2211	B
AR 2212	C
AR 2213	D
AR 2214	E
AR 2215	F

Подробности о режиме TERMINAL см. гл. 7.

#### 3.5.11 Счетчик отключений питания

AR 23 выдает информацию в виде 4-разрядного двоично-десятичного числа о числе пропаданий питания. Этот счетчик можно сбросить при необходимости операцией изменения текущего значения 1 с программатора (Подробности см. 7.2.4). Счетчик отключений питания обновляется каждый раз после включения питания.

#### 3.5.12 SYSMAC LINK - Флаги периферийного устройства

Периферийное устройство можно использовать системой SYSMAC LINK только с одного порта одновременно. При смене порта, с которого используется периферийное устройство, включите в 1 бит инициализации периферийного устройства SYSMAC LINK (AR 2403).

Бит	Функция
AR 2400	SYSMAC LINK - Флаг периферийного устройства RS-232C (Этот флаг включается в 1, когда периферийное устройство используется системой SYSMAC LINK через порт RS-232C)
AR 2401	SYSMAC LINK - Флаг периферийного устройства Порта А (Этот флаг включается в 1, когда периферийное устройство используется системой SYSMAC LINK через порт А панели связи)
AR 2402	SYSMAC LINK - Флаг периферийного устройства Порта В (Этот флаг включается в 1, когда периферийное устройство используется системой SYSMAC LINK через порт В панели связи)
AR 2403	SYSMAC LINK - Бит инициализации периферийного устройства (Включите этот бит в 1 для инициализации использования периферийного устройства системой SYSMAC LINK)

#### 3.5.13 Флаг времени цикла

AR 2405 включается в 1, когда время цикла, установленное командой SCAN короче, чем текущее время цикла.

AR 2405 обновляется каждый цикл, когда ПК находится в режиме RUN или PROGRAM.

### 3.5.14 Флаги установленных модулей связи

Следующие флаги указывают, когда модули связи установлены на панели (модули HOST LINK, установленные на ЦПУ, см. 3.5.15). Флаги обновляются каждый цикл.

Название	Бит	Модуль связи
Флаг установленного модулей связи SYSMAC LINK/ SYSMAC NET уровня 1	AR 2408	Модуль связи SYSMAC LINK/ SYSMAC NET уровня 1
Флаг установленного модулей связи SYSMAC LINK/ SYSMAC NET уровня 0	AR 2409	Модуль связи SYSMAC LINK/ SYSMAC NET уровня 0
Модуль HOST LINK уровня 1, установленный на панели	AR 2413	Модуль HOST LINK уровня 1, установленный на панели
Модуль HOST LINK уровня 0, установленный на панели	AR 2414	Модуль HOST LINK уровня 0, установленный на панели

### 3.5.15 Флаг устройства, установленного на ЦПУ

AR 2415 включается в 1, когда есть устройство, смонтированное непосредственно на ЦПУ. Среди них могут быть модули HOST LINK, программаторы и модули интерфейса. Этот флаг обновляется каждый цикл.

### 3.5.16 Бит переключения FPD

AR 2508 используются для настройки автоматического времени просмотра FPD. Подробности см. 5.25.12.

### 3.5.17 Флаги трассировки данных и биты управления

Следующие биты управления и флаги используются при трассировке данных командой TRSM. Во время операции трассировки флаг должен быть = 1. Флаг завершения установится в 1, когда оттрассировано достаточно данных для заполнения памяти трассировки.

Бит	Название
AR 2512	Флаг завершения трассировки
AR 2513	Флаг трассировки
AR 2514	Бит переключения трасс (можно писать)
AR 2515	Бит начала выборки (можно писать)

**Замечание** Подробности см. 5.25.3.

### 3.5.18 Индикаторы времени цикла

В AR 26 содержится время максимального цикла с начала исполнения программы. В AR 27 содержится время текущего цикла.

Оба времени даны в 0.1 мс 4-разрядным двоично-десятичным числом (000.0..999.9 мс) и обновляются каждый цикл.

## 3.6 Область DM (Data Memory)

Область DM делится на части, как показано в следующей таблице. Часть UM ( до 3 000 слов через каждые 1000 слов ) может быть выделена в качестве расширенной DM.

Адреса	Чтение/запись пользователя	Использование
DM 0000..DM 0999	Чтение/запись	Нормальные DM
DM 1000..DM 2599		Область модулей специальных входов/выходов <sup>1</sup>



Адреса	Чтение/запись пользователя	Использование
DM 2600..DM 5999		Нормальные DM
DM 6000..DM 6030		Протокол ошибок
DM 6100..DM 6143		Резерв
DM 6144..DM 6599	Чтение	Системные параметры
DM 6600..DM 6655		Установочные параметры ПК
DM 7000..DM 9999		Расширенные DM <sup>2</sup>

(1) - Установочными параметрами ПК можно задать использование DM 7000..DM 8599 в качестве специальной области входов/выходов вместо DM 1000..DM 2599. Подробности см. 3.6.4.

(2) - Операцию программатора выделения области UM можно использовать для выделения до 3 000 слов области UM в качестве дополнительных DM.

Хотя данные в области DM, подобно другим областям данных, состоят из 16-битовых слов, у них нельзя указать определенный бит в качестве операнда. DM 0000..DM 6143 можно записывать из программы, DM 6144..DM 6655 можно записывать только с периферийного устройства (программатора или компьютера с ПО).

Область DM сохраняет состояние при прерывании питания.

#### Косвенная адресация

Как правило, когда содержимое слова из области данных задано для команды, команда работает непосредственно с содержимым слова. Например, допустим, что команда MOV работает с DM 0100 как первым операндом и LR 20 как вторым операндом. При выполнении команды содержимое DM 0100 перешлется в LR 20.

**Замечание** Расширенные DM нельзя использовать для косвенной адресации.

Однако возможно использовать косвенные адреса в качестве операндов для многих команд. Для указания косвенного адреса перед адресом операнда ставится \*DM. Когда задана косвенная адресация, содержимое операнда содержит не данные, с которыми будет проводиться операция. Вместо этого там содержится адрес другого слова DM, в котором содержатся данные, которые будут использоваться в команде. Если \*DM 0100 используется в следующем примере, а содержимое DM 0100 = 0324, тогда \*DM 0100 фактически означает, что в качестве операнда следует использовать содержимое DM 0324, и содержимое DM 0324 будет передано в LR 00.

#### 3.6.1 Область расширенных DM

Область расширенных DM служит для предоставления памяти для рабочих параметров и других данных для модулей связи и модулей специальных входов/выходов. До 3 000 слов области UM (с наращиванием в 1 К слов) можно выделить с помощью операции UM. Адреса области расширенных DM - это DM 7000..DM 9999.

Данные в области расширенных DM можно передать в область по умолчанию для модулей специальных входов/выходов (DM 1000..DM 1999) при пуске ПК или командой программирования для замены параметров, что позволяет быстрое переключение между процессами управления. Область расширенных DM можно также использовать для сохранения параметров других устройств системы ПК, например, строк символов или цифровой таблицы программируемых терминалов.

Область расширенных DM используется для сохранения оперативных параметров и ее нельзя программировать как нормальную область DM. Область расширенных DM можно переписать только с периферийного устройства, она сохраняет состояние при пропадании питания и ее нельзя использовать для косвенной адресации.

Область UM может быть выделена для расширенных UM с наращиванием 1 К слов. После создания UM она сохраняется и передается как часть программы, т.е. для сохранения и передачи специальных процедур не требуются.

**Операция UM ALLOCATION (Выделение UM)**

Далее показана процедура проведения операции ВЫДЕЛЕНИЕ UM с программатора. Подробности о командах ОЧИСТКА ДАННЫХ и ВЫДЕЛЕНИЕ UM см. 4.6.3.

1, 2, 3,... 1. Очистите память.



**Замечание** ВЫДЕЛЕНИЕ UM невозможно без предварительной очистки памяти.

2. Область расширенных DM можно задать в 0, 1, 2 и 3 К слова. Следующая последовательность клавиш создает область расширенных DM в 2 К слов (DM 7000..DM 8999).



Нажмите клавишу 0 для убирания области расширенных DM (0 К слов), или

Нажмите клавишу 1 для выделения DM 7000.... DM 7999 (1 К слов) или

Нажмите клавишу 2 для выделения DM 7000.... DM 8999 (2 К слов) или

Нажмите клавишу 3 для выделения DM 7000.... DM 9999 (3 К слов).

**3.6.2 Область модулей специальных входов/выходов**

Модулям специальных входов/выходов выделены слова 1000..1600 в области DM в зависимости от значений в установочных параметрах DM 6602. Значение DM 6602 определяет, задается ли область данных для модулей специальных входов/выходов 10..16 и сохраняются ли DM 1000..DM 2599 в режиме читать/писать или только читать (DM 7000..DM 8599). Подробности см. Приложение E.

Модуль	Адреса
0	DM 1000..DM 1099 или DM 7000..DM 7099
1	DM 1100..DM 1199 или DM 7100..DM 7199
2	DM 1200..DM 1299 или DM 7200..DM 7299
3	DM 1300..DM 1399 или DM 7300..DM 7399
4	DM 1400..DM 1499 или DM 7400..DM 7499
5	DM 1500..DM 1599 или DM 7500..DM 7599
6	DM 1600..DM 1699 или DM 7600..DM 7699
7	DM 1700..DM 1799 или DM 7700..DM 7799
8	DM 1800..DM 1899 или DM 7800..DM 7899
9	DM 1900..DM 1999 или DM 7900..DM 7999
A	DM 2000..DM 2099 или DM 8000..DM 8099
B	DM 2100..DM 2199 или DM 8100..DM 8199
C	DM 2200..DM 2299 или DM 8200..DM 8299
D	DM 2300..DM 2399 или DM 8300..DM 8399
E	DM 2400..DM 2499 или DM 8400..DM 8499
F	DM 2500..DM 2599 или DM 8500..DM 8599

**Замечание** Эти DM можно использовать для других целей, если они не выделены для модулей специальных входов/выходов.

**3.6.3 Специальные Модули**

В следующей таблице перечислены Специальные Модули, которые можно использовать, и номера Модулей, которые можно применять для них.



### 3.12 Область EM (Расширенная область DM)

Наименование	Номер модели	Диапазон номеров Модуля
Модуль управления движением	C200H-MC221	0..8, A..E
Модуль позиционирования по кулачкам	C200H-CP114	0..9
Консоль доступа к данным	C200H-DSC01	
Модуль ASCII	C200H-ASC02	
Модуль логики FUZZY	C200H-FZ001	
Модуль голоса	C200H-OV001	
Модуль датчика ID	C200H-IDS01-V1	
	C200H-IDS21	

- Замечание**
1. Номера A..F можно присвоить только Модулям C200HX-CPU54/CPU64-ZE или C200HG-CPU53/63-ZE, изготовленным в или позже января 1996 (номер \*\*16). Другим Модулям ЦПУ можно присваивать номера только 0..9.
  2. Номера A..F можно присвоить только Модулям C200HX-CPU5\_/CPU6\_-ZE или C200HG-CPU5\_/6\_-ZE. Другим Модулям ЦПУ можно присваивать номера только 0..9.
  3. Номера A..E можно присвоить только Модулям C200HX-CPU54/CPU64-ZE или C200HG-CPU53/63-ZE, изготовленным в или позже декабря 1995 (номер \*\*16). Другим Модулям ЦПУ можно присваивать номера только 0..9.
  4. Номера 0..F можно присвоить только C200HX/HG/HE.
  5. Номера также должны быть присвоены Модулям связи, и Модули связи считаются Специальными Модулями, но слов им на выделяется.
  6. Модулям высокоскоростных счетчиков, Модулям позиционирования и модулям аналоговых входов/выходов выделяются слова в области DM. Другим Модулям слова в области DM не выделяются.
  7. Модулям C200H-NC221 и C200H-MC221 выделяются слова для двух специальных Модулей (т.е. для двух Модулей).

#### 3.6.4 Область протокола ошибок

DM 6000..DM 6030 используются для сохранения до 10 записей о характере, времени и дате ошибок, произошедших в ПК.

Область протоколов ошибок будет сохранять коды системных ошибок или вызванных FAL/FALS, когда AR 0715 (бит разрешения протокола) = 1.

Подробности о кодах ошибок см. гл. 9.

#### Структура области

Запись ошибки занимает 3 слова каждая среди слов DM 6001..DM 6030. Последнюю запись можно вызвать с помощью DM 6000 (указатель записи в протоколе). Номер записи, слова DM и значение указателя для каждой из 10 ошибок представляют собой следующее:

Запись	Адреса	Значения указателя
Нет	N.A. (нет адреса)	0000
1	DM 6001..DM 6003	0001
2	DM 6004..DM 6006	0002
3	DM 6007..DM 6009	0003
4	DM 6010..DM 6012	0004
5	DM 6013..DM 6015	0005
6	DM 6016..DM 6018	0006
7	DM 6019..DM 6021	0007
8	DM 6022..DM 6024	0008
9	DM 6025..DM 6027	0009

### 3.12 Область EM (Расширенная область DM)

10	DM 6028..DM 6030	000A
----	------------------	------

Хотя каждая из них содержит разные записи, структура каждой записи одинакова:

первое слово - код ошибки;

второе и третье слова - день и время.

Код ошибки может быть системным или от FAL, FALS. Время и день будут из AR 18 и AR 19 (область календарь/часы). Кроме того, в коде записывается, фатальная или нет (00) ошибка. Данная структура показана далее в таблице.

Слово	Бит	Содержание
Первое	00..07	Код ошибки
	08..15	00 (нефатальная) или 08 (фатальная)
Второе	00..07	Минуты
	08..15	Секунды
Третье	00..07	Часы
	08..15	День месяца

В таблице показаны возможные коды ошибок и соответствующие ошибки.

Серьезность ошибки	Код ошибки	Ошибка
Фатальные ошибки	00..99 или 9F	Системная ошибка (FALS)
	C0..C2	Ошибка шины входов/выходов
	E0	Ошибка ввода/вывода таблицы входов/выходов
	E1	Слишком много модулей
	F0	Нет команды END
	F1	Ошибка памяти
Нефатальные ошибки	00..99	Системная ошибка (FAL)
	8A	Ошибка входного прерывания
	8B	Ошибка программы прерываний
	9A	Ошибка входов/выходов высокой плотности группы 2
	9B	Ошибка установочных параметров ПК
	9C	Ошибка панели связи
	9D	Ошибка связи с кассетой памяти UM
	B0..B1	Ошибка удаленных входов/выходов
	D0	Ошибка специальных входов/выходов
	E7	Ошибка сверки таблицы входов/выходов
	F7	Ошибка аккумулятора
	F8	Превышение времени цикла

#### Работа

Когда появляется первый код ошибки с включенным AR 0715 в 1 (бит разрешения протокола ошибок), соответствующие данные будут занесены в записи протокола после записи, на которую указывал указатель, и указатель будет инкрементирован. Коды, появившиеся позже, будут помещены в следующие записи до заполнения последней. Обработка последующих записей об ошибках зависит от состояния AR 0713 (бит перезаписи протокола ошибок).

Если AR 0713 = 1 и указатель имеет значение 000A, следующая ошибка будет записана в запись 10, содержимое записи 10 будет перенесено в запись 9 и так далее, пока содержимое записи 1 не будет потеряно, т.е. область действует как сдвиговый регистр. Указатель записей будет оставаться на 000A.

Если AR 0713 = 0 и указатель имеет значение 000A, содержимое протокола будет оставаться неизменным и последующие коды записаны не будут, пока AR 0713 не будет установлена в 0 или не будет сброшена область протоколов ошибок.

Область протоколов ошибок можно сбросить путем включения в 1 и затем 0 бита сброса протокола (AR 0714). После этого указатель записи будет установлен на 0000, протокол будет очищен, все последующие коды будут записываться с начала области. AR 0715 (бит разрешения протокола) должен быть установлен в 1 для сброса области протокола ошибок.

#### 3.6.5 Установочные параметры ПК

Установочные параметры ПК (DM 6600..DM 6655) содержат параметры, которые определяют работу ПК. Установочные параметры ПК можно изменять с программатора или ПО, если UM не защищена от записи секцией 1 DIP ЦПУ. Подробности о переключателе DIP см. 2.5.

Параметры DM 6600..DM 6634 можно занести или изменить только в режиме ПК PROGRAM. Параметры DM 6635..DM 6655 можно занести или изменить в режиме ПК RUN или MONITOR. Следующие слова можно изменять с помощью меню ПО. (ПК должен находиться в режиме PROGRAM.)

- 1, 2, 3,... 1. Режим пуска (DM 6600)
2. Источник режима пуска (DM 6601)
3. Время контроля цикла (DM 6618)
4. Время цикла (DM 6619)
5. Параметры порта RS-232 C (DM 6645.. DM 6649)
6. Параметры периферийного порта (DM 6650.. DM 6654)

ПК может работать с установочными параметрами по умолчанию, изменять которые нужно только при подстройке ПК к условиям работы при решении специальных задач. Параметры ПК представлены в таблице.

Слово(а)	Бит(ы)	Функция	По умолчанию
<b>Обработка пуска (DM 6600..DM 6612)</b> Следующие параметры доступны только один раз после включения ПК.			
DM 6600	00..07	Режим пуска (действует, когда биты 08..15 = 02) 00: PROGRAM 01: MONITOR 02: RUN	00
	08..15	Источник режима ПУСК 00: Переключатель программатора 01: Продолжить работу в режиме, который был перед отключением питания 02: Из битов 00..07 данного параметра.	00
DM 6601	00..07	Резерв	-
	08..11	Состояние бита сохранения IOM (SR 25212) 0: Сбросить 1: Сохранить	0
	12..15	Состояние бита удержания принудительной установки (SR 25211) 0: Сбросить 1: Сохранить	0
DM 6602	00..07	Резерв	-

### 3.12 Область EM (Расширенная область DM)

Слово(а)	Бит(ы)	Функция	По умолчанию
	08..15	Область модулей специальных входов/выходов (Подробности об установочных параметрах специальных входов/выходов см. 3.4.8) 00: Режим RAM C200H-совместимый Использует DM 1000..DM 2599 для области специальных входов/выходов 01: Режим 1 ROM, C200H-совместимый Содержимое DM 7000..DM 7999 передается в DM 1000..DM 1999 при пуске и использует DM 1000..DM 1999. 02: Линейный Режим 1 DM DM 7000..DM 7999 используются для области модулей специальных входов/выходов 11: Режим 2 ROM, C200H-совместимый Содержимое DM 7000..DM 8599 передается в DM 1000..DM 2599 при пуске и использует DM 1000..DM 2599. 11: Линейный Режим 2 DM DM 7000..DM 8599 используются для области модулей специальных входов/выходов	00
DM 6603.. DM 6604	00..15	Не используется	-
DM 6605	00..07	Время мгновенного прерывания питания (0..10 мс) Задаёт Время мгновенного прерывания питания 00..10 (двоично-десятичное)	0 мс
	08..15	Не используется	-
DM 6606.. DM 6612	00..15	Не используется	-
<b>Параметры связи и времени цикла (DM 6613..DM 6619)</b> Следующие параметры доступны только при пуске программы.			
DM 6613	00..07	Время обслуживания порта В панели связи (действуют, когда биты 08..15 = 01) 00..99:Процент от времени цикла на обслуживание порта В панели связи (двоично-десятичное число). Минимум: 0.228 с, максимум 58.254 мс.	Не установлен (0000)
	08..15	Разрешение задания времени обслуживания порта В панели связи 00: Время обслуживания не задается (фиксирован на 5%, 0.228 мс минимум) 01: время, заданное битами 00..07 Время обслуживания = 10 мс при остановке отработки программы, независимо от задания.	Не установлен (0000)
DM 6614	00..07	Время обслуживания порта А панели связи (действуют, когда биты 08..15 = 01) 00..99:Процент от времени цикла на обслуживание порта А панели связи (двоично-десятичное число). Минимум: 0.228 с, максимум 58.254 мс.	Не установлен (0000)
	08..15	Разрешение задания времени обслуживания порта А панели связи 00: Время обслуживания не задается (фиксирован на 5%, 0.228 мс минимум) 01: время, заданное битами 00..07 Время обслуживания = 10 мс при остановке отработки программы, независимо от задания.	Не установлен (0000)
DM 6615	00..15	Зарезервировано	-
DM 6616	00..07	Время обслуживания порта RS-232C (действуют, когда биты 08..15 = 01) 00..99:Процент от времени цикла на обслуживание порта RS-232C (двоично-десятичное число) Минимум: 0.228 с, максимум 58.254 мс	Не установлен (0000)

### 3.12 Область EM (Расширенная область DM)

Слово(а)	Бит(ы)	Функция	По умолчанию
	08..15	Разрешение задания времени обслуживания порта RS-232C 00: Время обслуживания не задается (фиксирован на 5%, 0.228 мс минимум) 01: Время, заданное битами 00..07 Время обслуживания = 10 мс при остановке отработки программы, независимо от задания.	Не установлен (0000)
DM 6617	00..07	Время обслуживания периферийного порта (действуют, когда биты 08..15 = 01) 00..99: Процент от времени цикла для обслуживания периферийного порта (двоично-десятичное число) Минимум: 0.228 с, максимум 58.254 мс	Не установлен (0000)
	08..15	Разрешение задания времени обслуживания периферийного порта 00: Время обслуживания не задается (фиксирован на 5%, 0.228 мс минимум) 01: время, заданное битами 00..07 Время обслуживания = 10 мс при остановке отработки программы, независимо от задания.	Не установлен (0000)
DM 6618	00..07	Время контроля за циклом (действуют, когда биты 08..15 = 01, 02 или 03) 00..99 (двоично-десятичные цифры) x дискрета задания (см. биты 08..15)	00
	08..15	Разрешение контроля за циклом (задание в битах 00..07 x дискрету; макс. 99 с) 00: 120 мс (задание бит 00..07 запретить) 01: Дискрета задания: 10 мс 02: Дискрета задания: 100 мс 03: Дискрета задания: 1 с	00: 120 мс
DM 6619	00..15	Время цикла 0000: Непостоянное (минимальное не задано) 0001..9999 (двоично-десятичное число): Минимальное время в мс	Непостоянно е
<b>Обработка прерываний/обновления (DM 6620..DM 6623)</b> Следующие параметры доступны только при пуске программы.			
DM 6620	00..09	Циклическое обновление модулей специальных входов/выходов (Номер бита соответствует номеру модулей, включая модули PC LINK) 0: Разрешить циклическое обновление и IOFR из главной программы 1: Запретить (обновление только командой IOFR из программ прерывания) Задание 01 (запретить) действует только когда реакция на прерывание установлена в режим высокоскоростной реакции. Для нормальной реакции на прерывания действует только для модулей специальных входов/выходов, смонтированных на ведомой панели.	Разрешить
	10..11	Зарезервировано	-
	12..15	Реакция на прерывания 0: Нормально (совместимо с C200H) Прерывания не могут быть приняты во время обслуживания HOST LINK, выполнения одной команды, обработки удаленных входов/выходов или специальных входов/выходов. Подпрограмма прерывания будет выполнена после завершения отработки. 1: Высокоскоростная реакция (C200HS или C200HX/G/HE) Прерывания будут приняты во время обслуживания HOST LINK, выполнения одной команды, обработки удаленных входов/выходов или специальных входов/выходов. Если это входное прерывание, текущая отработка будет прервана и будет выполнена подпрограмма прерывания.	Нормально
DM 6621	00..07	Зарезервировано	-



### 3.12 Область EM (Расширенная область DM)

Слово(а)	Бит(ы)	Функция	По умолчанию
	08..15	Обновление модулей специальных входов/выходов (включая модули PC LINK) 00: Разрешить обновление всех модулей специальных входов/выходов 01: Запретить обновление всех модулей специальных входов/выходов (не распространяется на ведомую панель) Задание 1 (запретить) не действует для модулей специальных входов/выходов, смонтированных на ведомых панелях	Разрешить
DM 6622	00..07	Дискрета времени прерывания по расписанию 00: 10 мс 01: 1 мс	10 мс (0000)
	08..15	Разрешение дискреты времени прерывания по расписанию 00: Запретить (10 мс) 01: Разрешить заданное в 00..07	10 мс (0000)
DM 6623	00..15	Циклическое обновление модулей специальных входов/выходов (включая модули PC LINK) (Номера битов соответствуют номерам модулей 0..F) 0: Разрешить циклическое обновление и IOFR из главной программы 1: Запретить (обновление только командой IOFR из программ прерываний) Задание 01 (запретить) действует только когда реакция на прерывание установлена в режим высокоскоростной реакции. Для нормальной реакции на прерывания действует только для модулей специальных входов/выходов, смонтированных на ведомой панели.	Разрешить
DM 6624.. DM 6644	00..15	Резерв	-
<b>Установочные параметры порта RS-232C</b> Следующие параметры доступны постоянно при включенном ПК			
DM 6645	00..03	Установочные параметры порта 00: Стандартные (1 стартовый бит, 7 бит данных, четн., 2 стоповых бита, 9 600 бод) 01: Задание в DM 6646	Стандартные
	04..07	Параметры управления CTS (сброс передатчика) 0: Разрешить управление CTS 0: Запретить управление CTS	Запретить
	08..11	Слова связи для 1:1 0: LR 00..LR 63 1: LR 00..LR 31 2: LR 00..LR 15. Максимальный номер узла ПТ для связи NT 1:N 1..7 (двоично-десятичные) (1..3 при C200HE-CPU__-E)	LR 00..LR 63
	12..15	Режим связи 0: Host Link 1: RS-232C 2: 1+1 (ведомый) 3: 1+1 (ведущий) 4: NT (1:1) 5: NT (1:N)	Host Link
DM 6646	00..07	Скорость обмена 00: 1.2 К 01: 2.4 К 02: 4.8 К 03: 9.6 К 04: 19.2 К	1.2 К

### 3.12 Область EM (Расширенная область DM)

Слово(а)	Бит(ы)	Функция	По умолчанию																																																																	
	08..15	<p>Формат кадра</p> <table border="1"> <thead> <tr> <th></th> <th>Старт</th> <th>Длина</th> <th>Стоп</th> <th>Четность</th> </tr> </thead> <tbody> <tr><td>00:</td><td>1 бит</td><td>7 бит</td><td>1 бит</td><td>четн.</td></tr> <tr><td>01:</td><td>1 бит</td><td>7 бит</td><td>1 бит</td><td>нечетн.</td></tr> <tr><td>02:</td><td>1 бит</td><td>7 бит</td><td>1 бит</td><td>нет</td></tr> <tr><td>03:</td><td>1 бит</td><td>7 бит</td><td>2 бита</td><td>четн.</td></tr> <tr><td>04:</td><td>1 бит</td><td>7 бит</td><td>2 бита</td><td>нечетн.</td></tr> <tr><td>05:</td><td>1 бит</td><td>7 бит</td><td>2 бита</td><td>нет</td></tr> <tr><td>06:</td><td>1 бит</td><td>8 бит</td><td>1 бит</td><td>четн.</td></tr> <tr><td>07:</td><td>1 бит</td><td>8 бит</td><td>1 бит</td><td>нечетн.</td></tr> <tr><td>08:</td><td>1 бит</td><td>8 бит</td><td>1 бит</td><td>нет</td></tr> <tr><td>09:</td><td>1 бит</td><td>8 бит</td><td>2 бит</td><td>ачетн.</td></tr> <tr><td>10:</td><td>1 бит</td><td>8 бит</td><td>2 бита</td><td>нечетн.</td></tr> <tr><td>11:</td><td>1 бит</td><td>8 бит</td><td>2 бита</td><td>нет</td></tr> </tbody> </table>		Старт	Длина	Стоп	Четность	00:	1 бит	7 бит	1 бит	четн.	01:	1 бит	7 бит	1 бит	нечетн.	02:	1 бит	7 бит	1 бит	нет	03:	1 бит	7 бит	2 бита	четн.	04:	1 бит	7 бит	2 бита	нечетн.	05:	1 бит	7 бит	2 бита	нет	06:	1 бит	8 бит	1 бит	четн.	07:	1 бит	8 бит	1 бит	нечетн.	08:	1 бит	8 бит	1 бит	нет	09:	1 бит	8 бит	2 бит	ачетн.	10:	1 бит	8 бит	2 бита	нечетн.	11:	1 бит	8 бит	2 бита	нет	00: 1 бит, 7 бит, 1 бит, четн.
	Старт	Длина	Стоп	Четность																																																																
00:	1 бит	7 бит	1 бит	четн.																																																																
01:	1 бит	7 бит	1 бит	нечетн.																																																																
02:	1 бит	7 бит	1 бит	нет																																																																
03:	1 бит	7 бит	2 бита	четн.																																																																
04:	1 бит	7 бит	2 бита	нечетн.																																																																
05:	1 бит	7 бит	2 бита	нет																																																																
06:	1 бит	8 бит	1 бит	четн.																																																																
07:	1 бит	8 бит	1 бит	нечетн.																																																																
08:	1 бит	8 бит	1 бит	нет																																																																
09:	1 бит	8 бит	2 бит	ачетн.																																																																
10:	1 бит	8 бит	2 бита	нечетн.																																																																
11:	1 бит	8 бит	2 бита	нет																																																																
DM 6647	00..15	<p>Задержка передачи 0000..9999 (двоично-десятичные цифры): дискрета = 10 мс</p>	0 мс																																																																	
DM 6648	00..07	<p>Номер узла (Host LINK) 00..31 (двоично-десятичные цифры)</p>	0																																																																	
	08..11	<p>Разрешение кода пуска (RS-232C) 0: Не разрешен 1: Задан</p>	Не разрешен																																																																	
	12..15	<p>Разрешение кода окончания (RS-232C) 0: Не разрешен (прием заданного числа байт) 1: Задан (задан код окончания) 2: CR, LF (возврат каретки, перевод строки)</p>	Не разрешен																																																																	
DM 6649	00..07	<p>Код старта (RS-232C ) 00..FF (двоичное число)</p>	Не использует-ся (0000)																																																																	
	08..15	<p>Биты 12..15 DM 6648 = 0: Количество принимаемых байт: 00: Значение по умолчанию (256 байт) 01..FF: 1..256 байт Биты 12..15 DM 6648 = 1: Код окончания (RS-232C) 00..FF (двоичное число)</p>																																																																		
<b>Установочные параметры периферийного порта (DM 6550..DM 6654)</b>																																																																				
Следующие параметры доступны постоянно при включенном ПК																																																																				
DM 6650	00..03	<p>Установочные параметры порта 00: Стандартные (1 стартовый бит, 7 бит данных, четн., 2 стоповых бита, 9 600 бод) 01: Задание в DM 6646</p>	Стандартные																																																																	
	04..07	<p>Параметры управления CTS (сброс передатчика) 0: Разрешить управление CTS 0: Запретить управление CTS</p>	Запретить																																																																	
	08..11	<p>Слова связи для 1:1 0: LR 00..LR 63 1: LR 00..LR 31 2: LR 00..LR 15. Максимальный номер узла ПТ для связи NT 1:N 1..7 (двоично-десятичные) (1..3 при C200HE-CPU__-E)</p>	LR 00..LR 63																																																																	
	12..15	<p>Режим связи 0: Host Link 1: RS-232C 2: 1+1 (ведомый) 3: 1+1 (ведущий) 4: NT (1:1) 5: NT (1:N)</p>	Host Link																																																																	

### 3.12 Область EM (Расширенная область DM)

Слово(а)	Бит(ы)	Функция	По умолчанию																																																																
DM 6651	00..07	Скорость обмена 00: 1.2 К 01: 2.4 К 02: 4.8 К 03: 9.6 К 04: 19.2 К	1.2 К																																																																
	08..15	Формат кадра  <table border="0"> <tr> <td></td> <td>Старт</td> <td>Длина</td> <td>Стоп</td> <td>Четность</td> </tr> <tr> <td>00:</td> <td>1 бит</td> <td>7 бит</td> <td>1 бит</td> <td>четн.</td> </tr> <tr> <td>01:</td> <td>1 бит</td> <td>7 бит</td> <td>1 бит</td> <td>нечетн.</td> </tr> <tr> <td>02:</td> <td>1 бит</td> <td>7 бит</td> <td>1 бит</td> <td>нет</td> </tr> <tr> <td>03:</td> <td>1 бит</td> <td>7 бит</td> <td>2 бита</td> <td>четн.</td> </tr> <tr> <td>04:</td> <td>1 бит</td> <td>7 бит</td> <td>2 бита</td> <td>нечетн.</td> </tr> <tr> <td>05:</td> <td>1 бит</td> <td>7 бит</td> <td>2 бита</td> <td>нет</td> </tr> <tr> <td>06:</td> <td>1 бит</td> <td>8 бит</td> <td>1 бит</td> <td>четн.</td> </tr> <tr> <td>07:</td> <td>1 бит</td> <td>8 бит</td> <td>1 бит</td> <td>нечетн.</td> </tr> <tr> <td>08:</td> <td>1 бит</td> <td>8 бит</td> <td>1 бит</td> <td>нет</td> </tr> <tr> <td>09:</td> <td>1 бит</td> <td>8 бит</td> <td>2 бит</td> <td>ачетн.</td> </tr> <tr> <td>10:</td> <td>1 бит</td> <td>8 бит</td> <td>2 бита</td> <td>нечетн.</td> </tr> <tr> <td>11:</td> <td>1 бит</td> <td>8 бит</td> <td>2 бита</td> <td>нет</td> </tr> </table>		Старт	Длина	Стоп	Четность	00:	1 бит	7 бит	1 бит	четн.	01:	1 бит	7 бит	1 бит	нечетн.	02:	1 бит	7 бит	1 бит	нет	03:	1 бит	7 бит	2 бита	четн.	04:	1 бит	7 бит	2 бита	нечетн.	05:	1 бит	7 бит	2 бита	нет	06:	1 бит	8 бит	1 бит	четн.	07:	1 бит	8 бит	1 бит	нечетн.	08:	1 бит	8 бит	1 бит	нет	09:	1 бит	8 бит	2 бит	ачетн.	10:	1 бит	8 бит	2 бита	нечетн.	11:	1 бит	8 бит	2 бита	нет
	Старт	Длина	Стоп	Четность																																																															
00:	1 бит	7 бит	1 бит	четн.																																																															
01:	1 бит	7 бит	1 бит	нечетн.																																																															
02:	1 бит	7 бит	1 бит	нет																																																															
03:	1 бит	7 бит	2 бита	четн.																																																															
04:	1 бит	7 бит	2 бита	нечетн.																																																															
05:	1 бит	7 бит	2 бита	нет																																																															
06:	1 бит	8 бит	1 бит	четн.																																																															
07:	1 бит	8 бит	1 бит	нечетн.																																																															
08:	1 бит	8 бит	1 бит	нет																																																															
09:	1 бит	8 бит	2 бит	ачетн.																																																															
10:	1 бит	8 бит	2 бита	нечетн.																																																															
11:	1 бит	8 бит	2 бита	нет																																																															
DM 6652	00..15	Задержка передачи 0000..9999 (двоично-десятичные цифры): дискрета = 10 мс	0 мс																																																																
DM 6653	00..07	Номер узла (Host LINK) 00..31 (двоично-десятичные цифры)	0																																																																
	08..11	Разрешение кода пуска (RS-232C) 0: Не разрешен 1: Задан	Не разрешен																																																																
	12..15	Разрешение кода окончания (RS-232C) 0: Не разрешен (прием заданного числа байт) 1: Задан (задан код окончания) 2: CR, LF (возврат каретки, перевод строки)	Не разрешен																																																																
DM 6654	00..07	Код старта (RS-232C ) 00..FF (двоичное число)	Не использует-ся (0000)																																																																
	08..15	Биты 12..15 DM 6648 = 0: Количество принимаемых байт: 00: Значение по умолчанию (256 байт) 01..FF: 1..256 байт  Биты 12..15 DM 6648 = 1: Код окончания (RS-232C) 00..FF (двоичное число)																																																																	
<b>Установочные параметры протокола ошибок (DM 6655)</b> Следующие параметры доступны постоянно при включенном ПК.																																																																			
DM 6655	00..03	Разрешение обнаружения ошибок программирования прерываний 0: Обнаруживать ошибки программирования прерываний 1: Не обнаруживать	Обнаружи-вать																																																																
	04..07	Резерв	-																																																																
	08..11	Разрешение контроля за временем цикла 0: Обнаруживать длинные циклы, как нефатальные ошибки 1: Не обнаруживать длинные циклы	Обнаружи-вать																																																																
	11..15	Разрешение предупреждения о неисправности аккумулятора 0: Обнаруживать падение напряжения аккумулятора как нефатальную ошибку 1: Не обнаруживать падение напряжения аккумулятора	Обнаружи-вать																																																																

#### 3.6.6 Установочные параметры панели связи

В DM 6550..DM 6554 содержатся установочные параметры панели связи порта В; В DM 6555...DM 6559 содержатся установочные параметры панели связи порта А.





### 3.12 Область EM (Расширенная область DM)

Слово(а)	Бит(ы)	Функция	По умолчанию
	12..15	Разрешение кода окончания (RS-232C) 0: Не разрешен (прием заданного числа байт) 1: Задан (задан код окончания) 2: CR, LF (возврат каретки, перевод строки)	Не разрешен
DM 6559	00..07	Код старта (RS-232C ) 00..FF (двоичное число)	Не использует-ся (0000)
	08..15	Биты 12..15 DM 6648 = 0: Количество принимаемых байт: 00: Значение по умолчанию (256 байт) 01..FF: 1..256 байт Биты 12..15 DM 6648 = 1: Код окончания (RS-232C) 00..FF (двоичное число)	

### 3.6.7 Установочные параметры области модулей специальных входов/выходов

Параметры в битах 08..15 слова DM 6602 определяют размер и место области модулей специальных входов/выходов, как показано в таблице.

Значение	Режим	Функция
00	C200H-совместимый Режим RAM	DM 1000..DM 2599 используются для области модулей специальных входов/выходов Данные в области модулей специальных входов/выходов можно переписать. Данные нельзя перевести в ROM.
01	C200H-совместимый Режим 1 ROM	Содержимое DM 7000..DM 7999 передается в DM 1000..DM 1999 при пуске и DM 1000..DM 1999 используются для области модулей специальных входов/выходов Предварительно нужно выполнить операцию выделения области UM. В C200H данный режим совместим с EEPROM и EPROM. Данные можно перевести в ROM непрямо, путем перевода в DM 7000..DM 7999
02	Режим 1 Линейных DM	DM 7000..DM 7999 используются для области модулей специальных входов/выходов DM 1000..DM 1999 можно использовать в качестве обычных DM. DM 7000..DM 7999 можно перевести в ROM.
11	C200H-совместимый Режим 2 ROM	Содержимое DM 7000..DM 8599 передается в DM 1000..DM 2599 при пуске и DM 1000..DM 2599 используются для области модулей специальных входов/выходов Предварительно нужно выполнить операцию выделения области UM. Данные можно перевести в ROM непрямо, путем перевода в DM 7000..DM 8599
12	Режим 2 Линейных DM	DM 7000..DM 8599 используются для области модулей специальных входов/выходов DM 1000..DM 2599 можно использовать в качестве обычных DM. DM 7000..DM 8599 можно перевести в ROM.

DM 7000..DM 9999 нельзя читать или переписывать прямо с программы. Для чтения этих данных из программы их нужно скопировать в другую область данных или область нормальных DM с помощью команды XD MR.

Когда установлен C200H-совместимый режим ROM, или Режим линейный DM, предварительно нужно выполнить операцию выделения области UM для выделения части памяти релейно-контактной программы для использования в качестве расширенных DM. Появится признак Системной ошибки (FAL 9B), если память не выделена для расширенных DM. Подробности об операции выделения области UM см. 7.2.15.

Когда установлен режим линейных DM, область данных модулей специальных входов/выходов будет начинаться с DM 7000, вместо DM 1000, так что добавьте 6000 к адресам DM, которые указаны в инструкции по работе модулей специальных входов/выходов.

Когда установочные параметры области данных модулей специальных входов/выходов = 01, 02, 11 или 12 и расширенные DM свыше DM 8000 не выделены, появится признак ошибки модулей специальных входов/выходов для модулей с номером A..F, когда Модуль обращается в свою выделенную область.

### 3.7 Область HR (Holding Relay)

Область HR используется для хранения/работы с различными данными и туда можно обращаться и к слову, и к биту. Адреса слов HR - это HR 00..HR 99, адреса битов - HR 0000..HR 9915. Биты HR можно использовать в любом порядке и программировать столько раз, сколько требуется.

Область HR сохраняет состояние при смене режимов работы, при прерывания питания или остановке отработки программы.

Биты и слова области HR можно использовать для сохранения данных при остановке отработки программы. Биты HR имеют также специальное применение, такое, как создание реле-фиксаторов командой KEEP и создание выходов самоподдержки. Они описаны в гл. 4. и гл. 5.

**Замечание** Требуемое число слов выделяются в диапазоне HR 00..HR 42 для таблиц маршрутизации и для просмотра таймеров при использовании систем SYSMAC NET.

### 3.8 Область TC (таймеров/ счетчиков)

Область TC используется для создания и программирования таймеров и счетчиков и содержит флаги завершения, заданные значения (SV) и текущие значения (PV) всех таймеров и счетчиков. Все доступны по номерам TC (TC 000..TC 511). Каждый номер TC задается как либо таймер, либо счетчик одной из следующих команд: TIM, TIMH, CNT, CNTR и TTIM. При использовании номера TC в команде таймера или счетчика префикс не требуется.

Когда номер TC задан одной из этих команд, его нельзя переопределить в другом месте программы этой же или другой командой. Если один и тот же номер TC уже определен в более чем одной из этих команд или в одной команде дважды, появится признак ошибки при проверке программы. На порядок использования номеров TC ограничений нет.

Если номер TC задан, его можно задавать как операнд в одной или более командах (за исключением ранее указанных). Когда номер TC задан как таймер, от используется с префиксом TIM. Префикс TIM используется независимо от команды, которой задавался таймер. Когда номер TC задан как счетчик, он используется с префиксом CNT. Префикс CNT также используется независимо от команды, которой задавался счетчик.

Номера TC можно назначать в качестве операндов, которые требуют битовых данных или словных данных. Если номер задан для данных типа бит, он будет обращаться к флагам завершения таймера/счетчика. Если номер TC задан для данных типа слово, он будет обращаться к ячейкам памяти, в которых хранится текущее значение (PV) таймера или счетчика.

Номера TC также используются для обращения к заданию (SV) таймеров или счетчиков с программатора. Процедуры использования программатора см. 7.1.

Область TC сохраняет задания (SV) таймеров и счетчиков при прерывания питания. Текущие значения (PV) таймеров сбрасываются при пуске отработки программы и при сбросе в секции INTERLOCK, но текущее значение счетчиков сохраняется. Подробности об операциях таймеров и счетчиков в секции INTERLOCK см. 5.10. В таком случае текущие значения счетчика не сбрасываются.

Обратите внимание, что в программировании "TIM 000" служит для обозначения трех вещей:

- задается команда таймера с TC = 000
- флаг завершения для данного таймера
- текущее значение (PV) данного таймера.

Значение контекста должно быть понятным, т.е. первая всегда команда, второй всегда бит, и третье всегда слово. То же верно для всех номеров TC с префиксами TIM или CNT.

### 3.9 Область LR (Link Relay)

Область LR используется в качестве области общих данных для передачи информации между ПК. Эта передача данных производится системой PC LINK.

Некоторые слова будут выделены в качестве слов для записи в каждом ПК. Эти слова пишутся из ПК и автоматически передаются в те же слова LR в другом ПК системы. Слова для записи другого ПК передаются в область чтения, так что каждый ПК может обратиться к данным, записанным другим ПК системы PC LINK. Только слова для записи, выделенные конкретному ПК, доступны для записи; все другие слова можно только читать. Подробности см. Инструкцию по системе PC LINK.

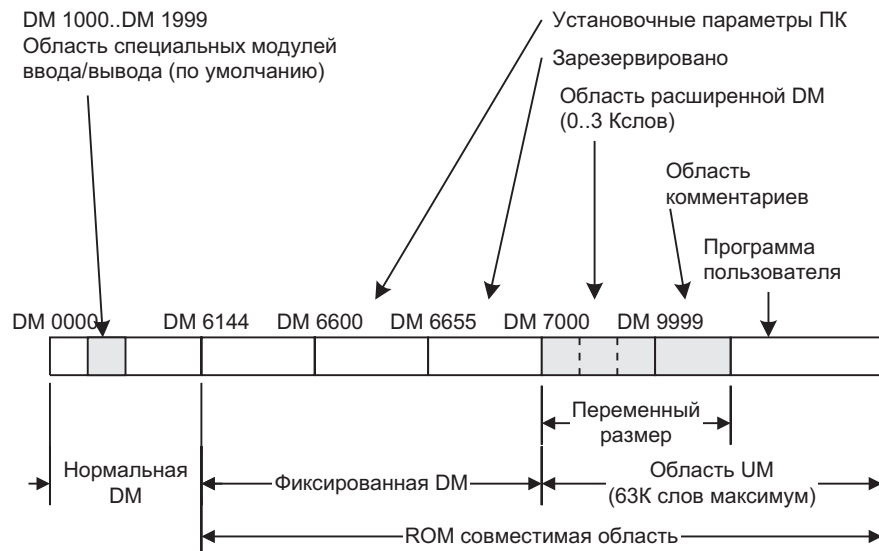
В области LR доступна обращение и к битам и к словам. Адреса слов области LR - это LR 00..LR 63. Адреса битов области LR - это LR 0000..LR 6315. Любая часть области LR, которая не используется системой PC LINK можно использовать в качестве рабочих бит или для систем SYSMAC NET или SYSMAC LINK.

Данные области LR не сохраняются при прерывания питания, когда ПК переключается в режим PROGRAM, или при сбросе в программной секции INTERLOCK. Подробности об INTERLOCK см. 5.10.

### 3.10 Область UM

В ПК C200HX/HG/HE область UM содержит релейно-контактную программу. Часть UM можно выделить для использования в качестве расширенных DM или области комментариев входов/выходов. Используемый размер области UM варьирует от 3.2 К слов в C200HE-CPU11-E(ZE) до 63.2 К слов в C200HX-CPU65/85-ZE.

Консоль программирования или ПО можно использовать для выделения расширенных DM, но область комментариев входов/выходов можно выделить только с помощью ПО. Структура областей DM и UM показана на рисунке.





**Замечание** Подробности об использовании ПО для выделения UM для расширенных DM или комментариев входов/выходов см. инструкцию по ПО. Подробности об использовании программатора для выделения UM для расширенных DM см. 7.2.15.

Область	Функция
Нормальные DM	Данную область можно использовать произвольно для команд расчетов и программирования. К DM можно обращаться только словами. DM 1000.. DM 2599 выделены модулям специальных входов/выходов (если они используются), но их можно использовать в качестве нормальных DM, когда область специальных входов/выходов заданы на DM 7000..DM 8599 в установочных параметрах ПК (DM 6602).
Установочные параметры ПК	Установочные параметры ПК содержат параметры для управления операциями ПК
Резерв	Данная область зарезервирована для системного пользования. Пользователю не доступна.
Расширенные DM	Эта область содержит данные инициализации, такие как данные модулей специальных входов/выходов, числа или строковые данные таблиц программируемого терминала и данные вычислений. Данные нельзя читать из области расширенных DM, как из простых DM. Расширенные DM можно переписать операцией модификации 16-ричных/двоично-десятичных данных с программатора или передав отредактированные данные из ПО.
Комментарии входов/выходов	Данная область служит для хранения комментариев входов/выходов, которые можно сохранить совместно с программой. Комментарии входов/выходов можно просмотреть без выполнения операции возвращения комментариев входов/выходов.
Релейно-контактная программа	Данная область служит для хранения релейно-контактной программы, созданной пользователем. Слова области UM, выделенные расширенным DM и/или области комментариев входов/выходов, забираются от области релейно-контактной программы

**Замечание**

1. Область релейно-контактной программы сокращается пропорционально, когда слова UM выделяются для расширенных DM и/или области комментариев входов/выходов. Перед выделением памяти для расширенных DM и/или области комментариев входов/выходов убедитесь, что достаточно памяти для релейно-контактной программы.
2. Установочные параметры по умолчанию для области UM не содержат выделения памяти для расширенных DM и/или области комментариев входов/выходов. Данная память выделяется пользователем по необходимости.

### 3.11 Область TR (временные ячейки)

В области TR есть 8 битов, которые используются только с командами LD и OUT для работы с некоторыми видами ветвлений при релейно-контактном программировании. Использование битов TR описано в гл. 4.

Адреса TR - это TR 0..TR 7. Каждый из этих битов можно использовать столько раз, сколько необходимо, и в любом порядке, если только тот же бит не используется дважды в одном модуле команд.

### 3.12 Область EM (Расширенная область DM)

В дополнение к области DM большой емкости, в ПК C200HG и C200HX есть область EM, в которой можно хранить до 96 К слов данных. Область EM делится на банки по 6 144 К слов каждый (EM 0000..EM 6143). В ПК C200HG есть один банк (банк 0), C200HX-CPU\_4-E(ZE) имеют три банка (банки 0, 1 и 2), C200HX-CPU65-ZE имеет восемь банков (банки 0..7), C200HX-CPU85-ZE имеет шестнадцать банков (банки 0..F). Текущий банк называется действующим.

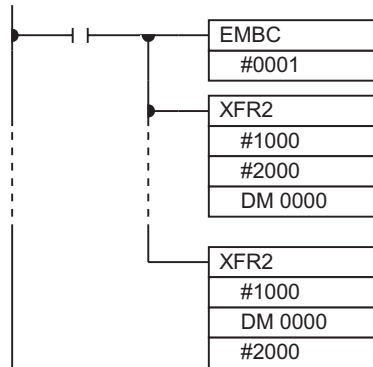
### 3.12.1 Использование области EM

К области EM нельзя обратиться прямо из большинства команд, но в ПК предусмотрены команды для управления данными области EM: EMBC, XFR2, BXF2 и IEMS.

Команда	Функция
EMBC	Заменяет текущий банк на заданный
XFR2	Переносит данные внутри текущего банка EM или между текущим банком EM и одной из нормальной областью данных
BXF2	Переносит данные между указанным банком EM и другим банком EM или нормальной областью данных
IEMS	Переключает адресата при косвенной адресации (*DM) на указанный банк EM. Может также переключать адресата назад на DM

#### Пример 1

Следующий пример использует команду EMBC для назначения текущего банка на банк 1 и XFR2 для передачи содержимого EM 2000..EM 2999 в DM 0000..DM 0999. После выполнения секции программы содержимое DM 000..DM 0999 передается обратно в EM 2000..EM 2999.

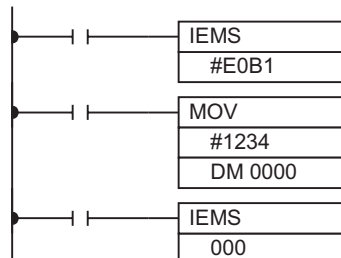


**Замечание** Если BXF2 использовался для передачи данных, можно задать любой банк EM и EMBC не потребуется для выбора банка 1.

#### Пример 2

Следующий пример использует команду IEMS для изменения адресата для косвенной адресации (\*DM) в EM банка 1. После выполнения данной команды операнды \*DM получают доступ в банк 1, а не в область DM. В данном случае второй операнд в команде MOV передает #1234 в слово банка EM. (Например, #1234 будет переслано в EM 0100, если в DM 0000 содержится 0100.

Далее в программе адрес для косвенной адресации (\*DM) переключится снова на область DM командой IEMS с операндом 000.



**Замечание**

1. При необходимости обязательно возвратите косвенную адресацию к ее значению по умолчанию (область DM). Адресат возвратится в область DM автоматически с началом следующего цикла.
2. Адресат косвенной адресации возвратится в область DM в начале подпрограмм прерываний, но может быть изменен внутри подпрограммы. Адресат возвратится к исходному заданию, когда управление возвратится в главную программу.

### 3.12.2 Текущий банк EM

Текущий банк EM настраивается на банк 0 при включении ПК и его можно изменить командами EMBC или IEMS. В отличие от адресата при косвенной адресации, номер текущего банка не инициализируется при начале цикла или при старте подпрограммы прерываний.2

После включения ПК произойдет возврат к переключенному состоянию банка после переключения режима ПК или завершения исполнения подпрограммы прерывания.



## **4. Написание и ввод программ**

*В данной главе дано описание базовых действий и понятий для написания основных релейно-контактных программ, ввода программы в память и ее исполнения. Приведены команды, которые используются для построения базовой структуры релейно-контактной схемы и управления ее исполнением. Полный набор команд описан в гл. 5.*

## 4.1 Основной алгоритм

Существует несколько основных операций при написании программы. Бланки, которые можно скопировать для помощи в программировании, приведены в Приложении F, Бланк присвоения слов и Приложении G, Бланк кодирования программы.

- 1, 2, 3,... 1. Сделайте список всех входных/выходных устройств и точек входов/выходов, которые им присвоены, и приготовьте таблицу, в которой показаны эти присвоения.
2. Если в ПК есть модули, которым выделены слова в областях, отличных от IR, или в выделенных словах IR функция каждого бита определяется типом модуля, приготовьте такие же таблицы, в которых будет показано, какие слова используются для модулей, и каково значение битов в слове. К этим модулям относятся специальные модули входы/выходов и модули связи.
3. Определите, какие слова можно использовать в качестве рабочих, и приготовьте таблицу, в которой будет показано их распределение.
4. Также приготовьте таблицу номеров ТС и номеров переходов, чтобы распределить их в соответствии с планом использования. Не забывайте, что функцию номера ТС можно задать только один раз в программе; Номера переходов 01..99 можно использовать один раз каждый. (Номер ТС описан в п. 5-14, переходы в том же пункте далее).
5. Нарисуйте релейно-контактную схему.
6. Введите программу в ЦПУ. При использовании программатора это потребует преобразование программы в мнемоническую форму.
7. Проверьте программу на синтаксические ошибки и исправьте их.
8. Выполните программу для проверки на ошибки исполнения и исправьте их.
9. После установки всей системы и готовности ее к работе выполните программу и в случае необходимости произведите точную наладку.
10. Сделайте резервную копию программы.

Основы релейно-контактного программирования и преобразования в мнемонакод описаны в п. 4.4. Приготовления для ввода программы с программатора описаны в 4.5.. 4.7.

Остальная часть главы 4 освещает более сложное программирование, меры предосторожности при программировании и исполнение программы. Все специальные применения команд описаны в гл. 5. Отладка описана в гл. 7. В гл. 9 также приведена информация, требуемая для отладки.

## 4.2 Терминология команд

При релейно-контактном программировании используется два базовых типа команд:

- команды, которые соответствуют условиям на РКС и используются в форме команд только для преобразования программы в мнемонакод;
- команды, которые располагаются с правой стороны релейно-контактной схемы и выполняются по результатам условий командных линий, ведущих к ним.

Большинство команд имеет минимум один или больше связанных с ними операндов.

Операнды указывают или содержат данные, с которыми должна исполняться команда.

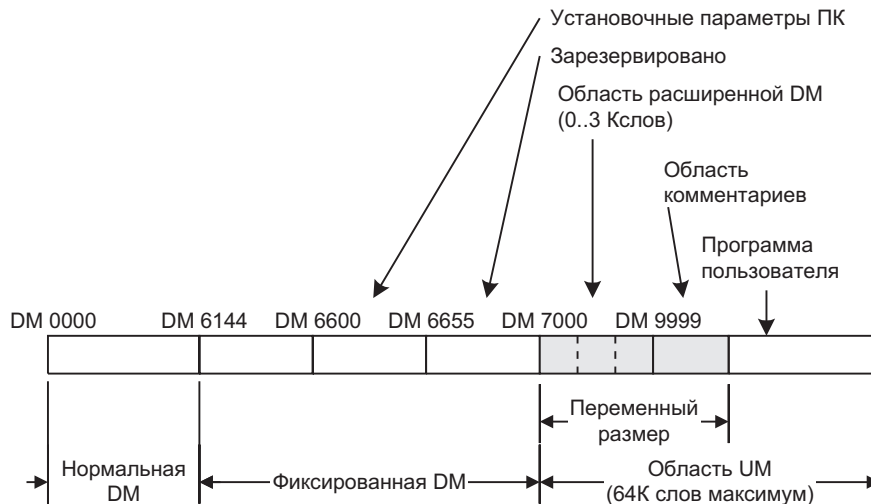
Операнды иногда вводятся как численные значения, но обычно представляют собой адреса слов или бит из области данных, в которых хранятся требуемые данные.

Например, команда MOVE, имеющая в качестве операнда источника IR 000, переместит содержимое IR 000 в заданное место. Данное место тоже задается как операнд. Бит, адрес которого задан как операнд, называется битовый операнд; слово, адрес которого задан как операнд, называется словный операнд; Если текущее значение введено как константа, ей предшествует # для указания того, что это не адрес.

Другие термины, применяемые для описания команд, приведены гл. 5.

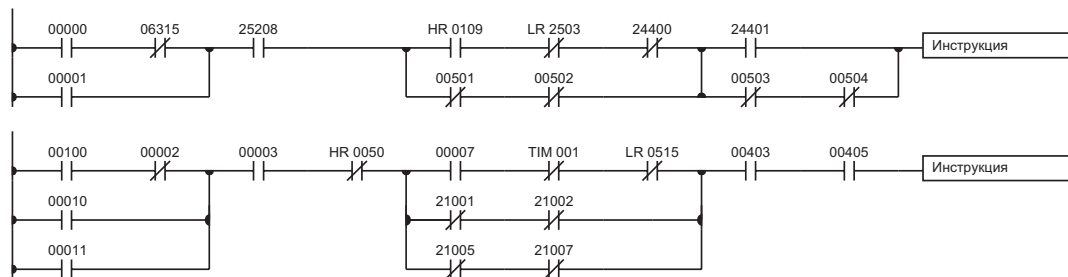
### 4.3 Объем памяти программ

Максимальный объем программы пользователя изменяется в зависимости от размера UM, выделенного для расширенных DM и зоны комментариев к входам / выходам. Примерно 10.1 К слов доступны для релейно-контактной программы, когда 3 К слов выделены для дополнительных DM и 2 К слов выделены для комментариев к входам / выходам, как показано ниже. Дальнейшую информацию о распределении UM см. 3.10.



### 4.4 Базовые релейно-контактные схемы

Релейно-контактная схема состоит из одной вертикальной линии с левой стороны с линиями, отходящими вправо. Вертикальная линия слева называется шиной, ответвления - командными линиями (или ступеньками). На командной линии располагаются условия, ведущие к "правосторонним" командам. Логические комбинации этих условий определяют, когда и как выполняются "правосторонние" команды. На рисунке приведен пример релейно-контактной схемы.



Как показано на диаграмме, командные линии могут разветвляться и снова соединяться. Две рядом расположенные вертикальные линии называются условием. Условие без диагональной черты называется нормально открытым условием и соответствует командам LOAD, AND или OR. Условие с диагональной чертой называется нормально закрытым условием и соответствует командам LOAD NOT, AND NOT или OR NOT. Число над каждым условием указывает битовый операнд команды. Это состояние бита определяет условие исполнения следующих команд. Как исполняется каждая команда в соответствии с условиями, описано далее. Перед рассмотрением этого, однако, следует объяснить базовые термины.

**Замечание** При индикации релейно-контактной схемы с помощью ПО с правой стороны будет индцироваться вторая шина, и она будет соединена со всеми "правосторонними" командами. Она совершенно не меняет релейно-контактную схему в функциональном плане. Между "правосторонними" командами и правой шиной нельзя поместить никакие условия, т.е. все "правосторонние" команды должны быть соединены

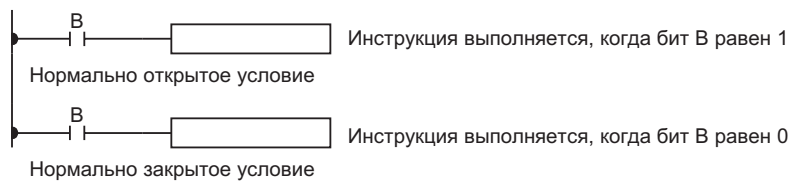
непосредственно с правой шиной. Подробности см. Инструкции по работе с ПО.

#### 4.4.1 Базовые термины

##### Нормально открытые и нормально закрытые условия

Каждое условие на РКС может находиться в состоянии либо 1, либо 0 в зависимости от состояния битового операнда, связанного с ним. Нормально открытое условие = 1, если битовый операнд = 1 и = 0, если битовый операнд = 0.

Нормально закрытое условие = 1, если битовый операнд = 0 и = 0, если битовый операнд = 1. То есть, Вы используете нормально открытое условие, если желаете выполнения команд, когда бит = 1, и нормально закрытое условие, если желаете выполнения команд, когда бит = 0.



##### Условия исполнения

В релейно-контактном программировании логическая комбинация условий 1 и 0 перед командой определяет общее условие, при котором она выполняется. Данное условие (либо 0, либо 1) называется условием исполнения команды. Все команды, за исключением команд LOAD, имеют условия исполнения.

##### Битовые операнды

Операндами для любой команды РКС могут быть любые биты в областях IR, SR, HR, AR, LR или TC. Это значит, что условия на релейно-контактной схеме могут определяться битами входов/выходов, флагами, рабочими битами, таймерами, счетчиками и т. д. Команды LOAD и OUTPUT могут также использовать биты области TR, но только в специальных прикладных задачах. Подробности см. 4.7.7.

##### Логические модули

Пути соответствия условий и команд определяются соотношением между условиями внутри командных линий, которые их соединяют. Любая группа условий, работающих совместно для выработки условия исполнения, называется логическим модулем. Хотя релейно-контактную схему можно составить без анализа отдельных логических модулей, понимание работы логических модулей необходимо для эффективного программирования и в случае, когда программы должны вводиться в виде мнемкода.

#### 4.4.2 Мнемокод

Релейно-контактную схему нельзя непосредственно ввести в ПК с программатора, требуется ПО. Для ввода с программатора необходимо преобразовать релейно-контактную схему в мнемокод. Мнемокод отображает ту же информацию, что и релейно-контактная схема, но в форме, которую можно ввести прямо в ПК. Вы можете программировать непосредственно в мнемокодах, но начинающим, или при сложных программах это не рекомендуется. Итак, независимо от того, используется ли программирующее устройство, программа загружается в ПК в мнемокоде. Поэтому важно понимать и мнемокод.

Из-за важности программатора как периферийного устройства и важности мнемкода в понимании программы, наряду с релейно-контактной схемой мы вводим и описываем мнемокоды. Помните, что при вводе с ПО знание мнемкода не требуется (хотя Вы, если захотите, можете их использовать с ПО).



### Структура памяти программ

Программа вводится по адресам памяти программ. Адреса в памяти программ несколько отличаются от адресов в других типов памяти, поскольку каждый адрес необязательно содержит одинаковый объем данных. Точнее, каждый адрес содержит одну команду и все определители и операнды (подробно описанные далее), требуемые для команды. Поскольку некоторые команды не требуют операндов, в то время как другие требуют до трех операндов, адреса памяти программ могут быть длиной 1..4 слова.

Адреса памяти программ начинаются с 00000 и продолжаются до тех пор, пока не будет занят весь объем памяти программ. Первое слово каждого адреса задает команду. Определители, используемые командой, также содержатся в первом слове. Кроме того, если команда требует только один битовый операнд (без определителя), битовый операнд также программируется в той же строке, что и команда. Остальные слова, требуемые командой, содержат операнды, которые определяют используемые данные. При преобразовании в мнемокод только релейно-контактные команды записываются в такой же форме, одно слово в строке, как они появляются в символах РКС. Пример мнемокодов приведен в таблице. Используемые команды описаны далее в инструкции.

Адрес	Инструкция	Операнд
00000	LD	HR 0001
00001	AND	00001
00002	OR	00002
00003	LD NOT	00100
00004	AND	00101
00005	AND LD	00102
00006	MOV	
		000
		DM 0000
00007	CMP	
		DM 0000
		HR 00
00008	LD	25505
00009	OUT	00501
00010	MOV	
		DM 0000
		DM 0500
00011	DIFU	00502
00012	AND	00005
00013	OUT	00503

Столбцы адрес и команда таблицы мнемокодов заполнен только для слов команд. Для всех других строк два левые столбца остаются незаполненными. Если команда не требует определителя или битового операнда, столбец операнд остается пустым для первой строки. Рекомендуем зачеркивать все незанятые ячейки столбцов (для всех слов команд, не требующих данных), чтобы быстро просмотреть столбец для проверки, не опущен ли какой-либо адрес.

При программировании адреса индицируются автоматически, и нет необходимости вводить адрес, если только по какой-либо причине нет необходимости поместить команду в каком-либо другом месте. При преобразовании в мнемокод лучше всего начинать с адреса памяти программ 00000, если нет особых причин размещать программу в другом месте.

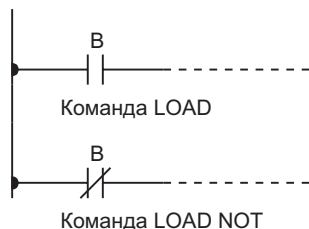
### 4.4.3 Команды РКС

Релейно-контактные команды - такие команды, которые соответствуют условиям на релейно-контактной схеме. Релейно-контактные команды, либо независимо, либо в

комбинации с блоковыми командами, описанными далее, образуют условия исполнения, в зависимости от которых исполняются все другие команды.

#### LOAD и LOAD NOT

Первыми условиями, которые начинают любой логический Модуль релейно-контактной схемы, являются команды LOAD и LOAD NOT. Эти команды занимают одну строку в мнемокоде. В следующем примере “Команда” может быть любой “правосторонней” командой, которые описаны далее в данной инструкции.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	Инструкция	
00002	LD NOT	00000
00003	Инструкция	

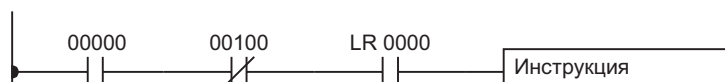
Когда данное условие является единственным в командной линии, условие исполнения “правосторонней” команды = 1 при данном условии = 1.

Для команды LOAD (нормально открытого условия) условие исполнения = 1 при IR 00000 = 1.

Для команды LOAD NOT (нормально закрытого условия) условие исполнения = 1 при IR 00000 = 0.

#### AND и AND NOT

Когда два или более условий расположены последовательно в одной командной линии, первое условие - LOAD или LOAD NOT. Остальные условия - AND и AND NOT. В примере показаны три условия, расположенные последовательно: LOAD, AND NOT и AND. Каждое из этих условий требует одной строки в мнемокоде.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND NOT	00100
00002	AND	LR 0000
00003	Инструкция	

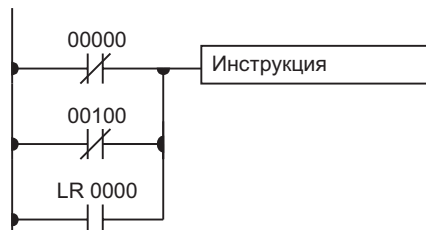
Условие исполнения команды будет = 1 только когда все три условия = 1, т.е. IR 0000 = 1, IR 00100 = 0, LR 0000 = 1.

Последовательно расположенные команды AND можно рассматривать индивидуально, каждая команда выполняет ЛОГИЧЕСКОЕ И над своим условием исполнения (т.е. результат всех условий до данной точки) и состоянием битового операнда самой команды AND. Если оба этих условия = 1, то условие исполнения для следующей команды будет = 1. Если хотя бы одно из этих условий = 0, то условие исполнения следующей команды будет = 0. Условие исполнения первой команды AND при последовательном соединении является первым условием командной линии.

Каждая команда AND NOT в последовательном соединении выполняет ЛОГИЧЕСКОЕ И над своим условием исполнения (т.е. результатом всех условий до данной точки) и инверсией своего битового операнда.

**OR и OR NOT**

Когда два или более условий расположены на разных командных линиях, идущих параллельно и затем соединяющихся, первое условие - LOAD или LOAD NOT. Остальные условия - команды OR и OR NOT. В примере показаны три условия, расположенные по порядку сверху: LOAD NOT, OR NOT и OR. Каждое из этих команд также требует одну строку в мнемокоде.



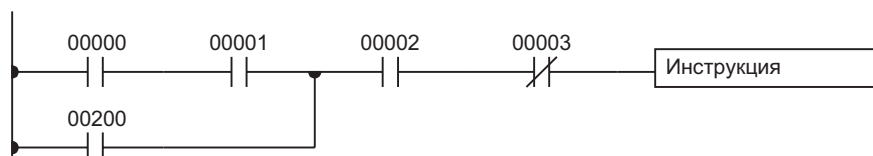
Адрес	Инструкция	Операнд
00000	LD NOT	00000
00001	OR NOT	00100
00002	OR	LR 0000
00003	Инструкция	

Условие исполнения для “правосторонней” команды будет = 1, когда хотя бы одно из условий = 1, т.е. когда IR 0000 = 0, IR 00100 = 0, или LR 0000 = 1.

Команды OR и OR NOT можно рассматривать индивидуально, каждая команда выполняет ЛОГИЧЕСКОЕ ИЛИ над своим условием исполнения и состоянием битового операнда самой команды OR. Если хотя бы одно из этих условий = 1, то результат исполнения следующей команды будет = 1.

**Сочетание команд AND и OR**

Когда команды AND и OR объединяются в сложные диаграммы, их иногда можно рассматривать индивидуально, каждая команда выполняет логическую операцию с предыдущим условием и состоянием бита операнда. Далее приведен пример такой ситуации. Изучите данный пример, пока не убедитесь, что мнемокод следует той же логике, что и релейно-контактная схема.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND	00001
00002	OR	00200
00003	AND	00002
00004	AND NOT	00003
00005	Инструкция	

В данном примере выполняется ЛОГИЧЕСКОЕ И между состоянием IR 0000 и IR 00001 для определения условия исполнения для ИЛИ с состоянием IR 0200. Результат данной операции определит условие исполнения для И с состоянием IR 00002, которое, в свою очередь, определит условие исполнения для И с инверсией (т.е. AND NOT) с состоянием IR 00003.

Однако в более сложных релейно-контактных схемах необходимо рассматривать логические модули перед определением условий исполнения для конечной команды, и именно здесь используются команды AND LOAD и OR LOAD. Перед рассмотрением более сложных диаграмм, мы рассмотрим требования команд для завершения простейшей программы “ввод/вывод”.

## 4.4.4 Вывод и Вывод инверсии (OUTPUT и OUTPUT NOT)

Простейший способ выдать результат комбинаций условий исполнения - прямая выдача командами OUTPUT и OUTPUT NOT. Данные команды используются для управления состоянием заданного битового операнда в соответствии со своим условием исполнения. Командой OUTPUT битовому операнду будет присваиваться значение 1, пока условие исполнения = 1 и 0, когда условие исполнения = 0.

Командой OUTPUT NOT битовому операнду будет присваиваться значение 1, пока условие исполнения = 0 и 0, когда условие исполнения = 1. Это происходит в соответствии с диаграммой. В мнемокоде каждая из этих команд требует одной строки.



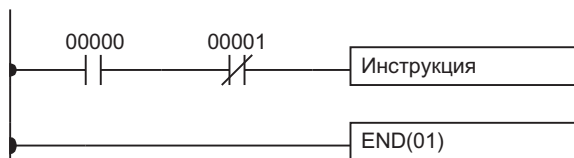
Адрес	Инструкция	Операнд
00000	LD	00000
00001	OUT	00200
00002	LD	00001
00003	OUT NOT	00201

В данном примере IR 00200 будет в положении 1, пока IR 00000 = 1, а IR 00201 будет = 0, пока IR 00001 = 1. В данном примере IR 00000 и IR 00001 - входные биты, а IR 00200 и IR 00201 - выходные биты, приписанные к модулям ПК. Т.е. сигналы, приходящие с входов, приписанных к IR 00000 и IR 00001, управляют выходами, приписанными к IR 00200 и IR 00201.

Время, которое биты, находятся в состоянии 1 или 0, можно задавать, комбинируя команды OUTPUT и OUTPUT NOT с командами таймера. Подробности см. 5-14-1.

## 4.4.5 Команда END

Последняя команда, необходимая для завершения простейшей программы - команда END. Когда ЦПУ исполняет программу, оно выполняет все команды до первой команды END, возвращается в начало программы и выполняет ее снова. Хотя команду END можно поместить в любую точку программы, что иногда делается при отладке, команды после первой встреченной команды END не будут исполняться, пока не будет удалена END. Число, следующее за командой END в мнемокоде - функциональный код, который используется для ввода большинства команд ПК. Об этом будут объяснения далее. Команда END не требует операндов и на командной линии END нельзя ставить никакую команду.



Адрес	Инструкция	Операнд
00500	LD	00000
00501	AND NOT	00001
00502	Инструкция	
00503	END	-

Если в программе нет команды END, программа вообще не будет выполняться.

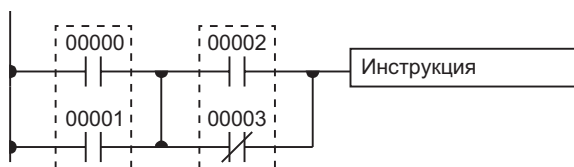
Теперь у Вас есть все команды, требующиеся для написания простейшей программы ввода/вывода. Перед тем, как покончить с основами релейно-контактных схем и перейти к вводу программ в ПК, давайте рассмотрим блочные AND LOAD и OR LOAD, поскольку они иногда необходимы в простых релейно-контактных схемах.

## 4.4.6 Блочные команды

Блочные команды не опрашивают конкретные условия релейно-контактной схемы, а описывают отношения между модулями. Команда AND LOAD производит операцию ЛОГИЧЕСКОЕ И над условиями исполнения, произведенными двумя логическими модулями. Команда OR LOAD производит операцию ЛОГИЧЕСКОЕ ИЛИ над условиями исполнения, произведенными двумя логическими модулями.

## AND LOAD

Хотя данная релейно-контактная схема проста по форме, она требует команды AND LOAD.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	OR	00001
00002	LD	00002
00003	OR NOT	00003
00004	AND LD	-

Два логических модуля отмечены пунктиром. На примере видно, что условие исполнение команды будет = 1,

когда какое-либо из условий левого модуля = 1 (т.е. либо IR 00000, либо IR 00001 = 1) и

когда какое-либо из условий правого модуля = 0 (т.е. либо IR 00002 = 1, либо IR 00003 = 0).

Вышеприведенную релейно-контактную схему нельзя преобразовать в мнемокод, используя только команды AND и OR. Если попытаться произвести ЛОГИЧЕСКОЕ И между IR 00002 и результатом ИЛИ между IR 00000 и IR 00001, теряется ИЛИ НЕ между IR 00002 и IR 00003 и ИЛИ НЕ прекращается, будучи ИЛИ НЕ между IR 00003 и результатом И между IR 00002 и первым ИЛИ. Нам же необходим метод произвести ИЛИ (NET) независимо и далее объединить результаты.

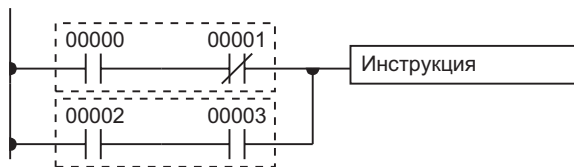
Для выполнения данной задачи мы пользуемся командой LOAD или LOAD NOT в середине командной строки. При таком выполнении LOAD или LOAD NOT текущее значение результатов операций сохраняется в специальных буферах, и логический процесс начинается снова. Для объединения результатов текущих условий исполнения с ранее “неиспользованными” условиями исполнения используется команда AND LOAD или OR LOAD. Здесь LOAD подразумевает загрузку последнего неиспользованного условия исполнения. Неиспользованное условие исполнения производится использованием команд LOAD или LOAD NOT для любого, кроме первого, условия в командной линии.

Анализируя приведенную релейно-контактную схему с точки зрения мнемокоманд, условием перед IR 00000 является команда LOAD, а условием после нее - ИЛИ между состоянием IR 00000 и IR 00001. Условием на IR 00002 является другая команда LOAD, а условием после нее является команда ИЛИ НЕ, т.е. ИЛИ между состоянием IR 00002 и инверсией IR 00003. Для получения условия исполнения “правосторонней” команды нужно проделать операцию И над этими двумя модулями. Это делает команда AND LOAD. Мнемокод для релейно-контактной схемы приведен в таблице. Команда AND LOAD не требует операндов, поскольку она работает с предварительно полученными условиями исполнения. Черточки в ячейке таблицы указывают на то, что не требуется операндов.

**OR LOAD**

Следующая диаграмма требует команды OR LOAD между верхним и нижним логическими модулями. Условие исполнения для “правосторонней” команды будет = 1, либо когда IR 00000 = 1 и IR 00001 = 0  
либо когда IR 00002 = 1 и IR 00003 = 1.

Задание в мнемокоде операции OR LOAD аналогично команде AND LOAD, за исключением того, что над текущим условием исполнения и последним неиспользованным условием исполнением производится операция ИЛИ.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND NOT	00001
00002	LD	00002
00003	AND	00003
00004	OR LD	

Разумеется, некоторые диаграммы могут потребовать обеих команд AND LOAD и OR LOAD.

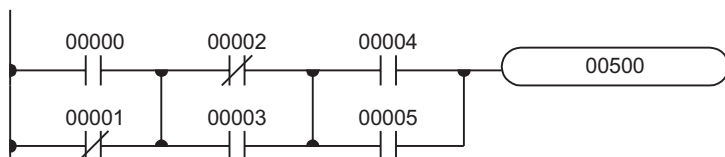
**Последовательное включение блоковых команд**

Для кодирования диаграмм с последовательным включением блоковых команд диаграмма должна быть разбита на логические модули. Каждый Модуль кодируется с использованием команды LOAD для кодирования первого условия, затем AND LOAD или OR LOAD для логического объединения модулей. И для AND LOAD, и для OR LOAD есть 2 метода.

1. Закодировать блоковую команду после первых двух модулей и далее после каждого нового модуля.
2. Кодирование всех модулей, подлежащих объединению, начиная каждый Модуль с LOAD или LOAD NOT, а затем блоковые команды для их объединения. В этом случае команды для последней пары модулей должны быть заданы первыми, и далее для каждого предшествующего модуля, возвращаясь к первому блоку.

Хотя оба этих метода производят одинаковый результат, второй метод может применяться только в случае, когда число модулей не более 8, т.е. требуется не более 7 блоковых команд.

Следующая диаграмма требует AND LOAD для преобразования в мнемокод, поскольку последовательно расположены три параллельных условия. Показаны два метода кодирования программы.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	OR NOT	00001
00002	LD NOT	00002
00003	OR	00003
00004	AND LD	-

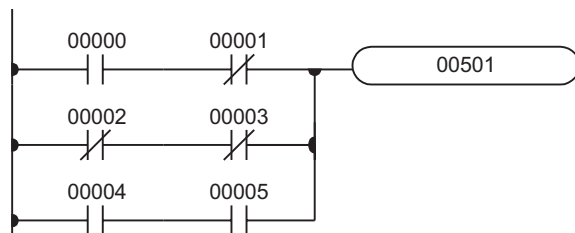
#### 4.4 Базовые релейно-контактные схемы

Адрес	Инструкция	Операнд
00005	LD	00004
00006	OR	00005
00007	AND LD	-
00008	OUT	00500

Адрес	Инструкция	Операнд
00000	LD	00000
00001	OR NOT	00001
00002	LD NOT	00002
00003	OR	00003
00004	LD	00004
00005	OR	00005
00006	AND LD	-
00007	AND LD	-
00008	OUT	00500

Методом, приведенным во второй таблице, можно объединить максимум 8 логических модулей. Для первого метода число логических модулей не ограничено.

Следующая диаграмма требует команд OR LOAD для преобразования в мнемокод, поскольку три пары последовательных цепочек расположены параллельно.



Первое из каждой пары условий преобразуется в LOAD с соответствующим битовым операндом и затем производится И с другим условием. Можно сначала закодировать первые 2 модуля, затем OR LOAD, затем последний Модуль и другая OR LOAD, или можно закодировать сначала три модуля, и далее 2 OR LOAD. Мнемокод обоих методов приведен в таблице.

Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND NOT	00001
00002	LD NOT	00002
00003	AND NOT	00003
00004	OR LD	-
00005	LD	00004
00006	AND	00005
00007	OR LD	-
00008	OUT	00501

Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND NOT	00001
00002	LD NOT	00002
00003	AND NOT	00003

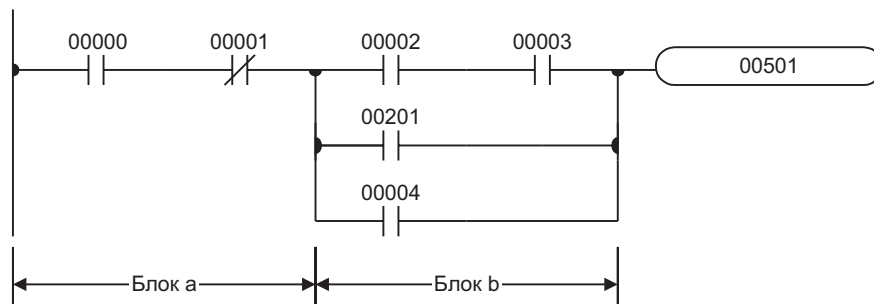
Адрес	Инструкция	Операнд
00004	LD	00004
00005	AND	00005
00006	OR LD	-
00007	OR LD	-
00008	OUT	00501

Методом, приведенным во второй таблице, можно объединить максимум 8 логических модулей. Для первого метода число логических модулей не ограничено.

**Комбинирование AND LOAD и OR LOAD**

Если число модулей не превышает 8, при использовании AND LOAD и OR LOAD можно использовать оба вышеописанных метода.

Следующая диаграмма содержит только два логических модуля. Нет необходимости дальнейшего дробления компонентов модулей, поскольку их можно закодировать простыми AND и OR.

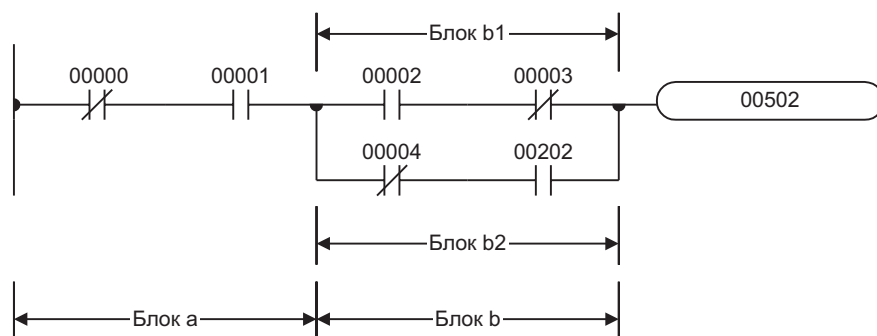


Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND NOT	00001
00002	LD	00002
00003	AND	00003
00004	OR	00201
00005	OR	00004
00006	AND LD	-
00007	OUT	00501

Хотя следующая диаграмма похожа на предыдущую, Модуль б нельзя закодировать, не разделив его на 2 модуля, объединенных OR LOAD. В данном примере сначала закодированы три модуля и далее применена команда OR LOAD для объединения двух последних модулей, а потом AND LOAD для объединения условия исполнения, полученного командой AND LOAD с условием исполнения модуля а.

При кодировании блоковых команд в конце логических модулей, которые они объединяют, они должны кодироваться в обратном порядке, т.е. вначале идет блоковая команда для последних двух модулей, далее команда, объединяющая условие исполнения, полученное первой блоковой командой, и условие исполнения модуля, третьего от конца и снова первый Модуль.





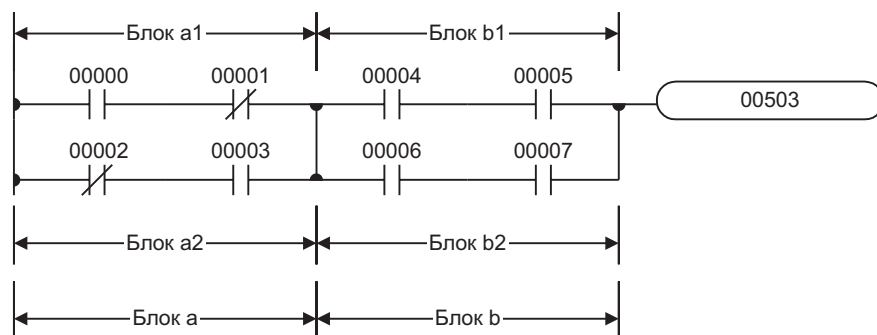
Адрес	Инструкция	Операнд
00000	LD NOT	00000
00001	AND	00001
00002	LD	00002
00003	AND NOT	00003
00004	LD NOT	00004
00005	AND	00202
00006	OR LD	-
00007	AND LD	-
00008	OUT	00502

### Сложные диаграммы

При определении, какие блоковые команды будут необходимы для кодирования диаграмм, иногда необходимо разбить диаграмму на большие модули и затем продолжать дробить большие модули до тех пор, пока не будут сформированы модули, которые можно программировать без блоковых команд. Затем данные модули кодируются, сначала объединяются маленькие модули, затем объединяются большие. Для объединения модулей используются команды AND LOAD или OR LOAD, т.е. AND LOAD или OR LOAD всегда объединяют два последних существующих условия исполнения, независимо от того, является ли оно результатом от единственного условия, логического модуля или предыдущих блоковых команд.

При работе со сложными диаграммами модули начинают кодировать с левого верхнего края, двигаясь сначала вниз, затем пересечение вправо. Это значит, что, если имеется выбор, OR LOAD нужно кодировать перед AND LOAD.

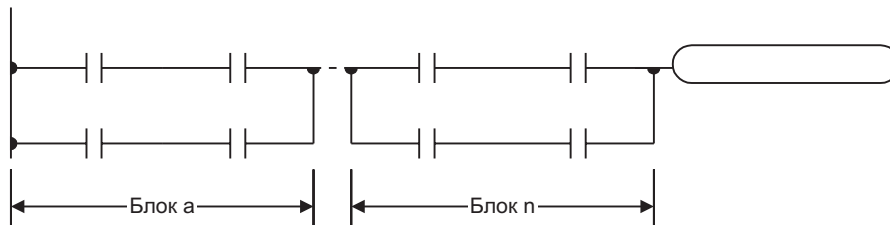
Следующую диаграмму перед тем, как ее можно будет кодировать, нужно разбить на 2 модуля и каждый из этих модулей разбить на 2 модуля. Как показано, модули а и b требуют AND LOAD. Перед применением AND LOAD нужно применить OR LOAD для объединения верхних и нижних модулей с каждой стороны, т.е. для объединения а1 и а2; b1 и b2.



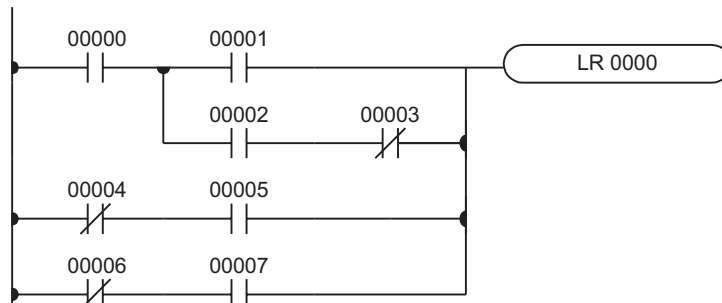
Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND NOT	00001

Адрес	Инструкция	Операнд
00002	LD NOT	00002
00003	AND	00003
00004	OR LD	-(блок a1 и a2)
00005	LD	00004
00006	AND	00005
00007	LD	00006
00008	AND	00007
00009	OR LD	-(блок b1 и b2)
00010	AND LD	-(блок a и b)
00011	OUT	00503

Следующий тип диаграммы можно легко закодировать, если каждый Модуль кодировать по порядку: сначала сверху вниз, затем слева направо. В следующей диаграмме модули a и b объединяются командой AND LOAD, затем кодируется Модуль c и объединяется второй командой AND LOAD с условием исполнения от предыдущего AND LOAD. Затем кодируется Модуль d и объединяется третьей командой AND LOAD с условием исполнения от второго AND LOAD, и так далее до модуля n.



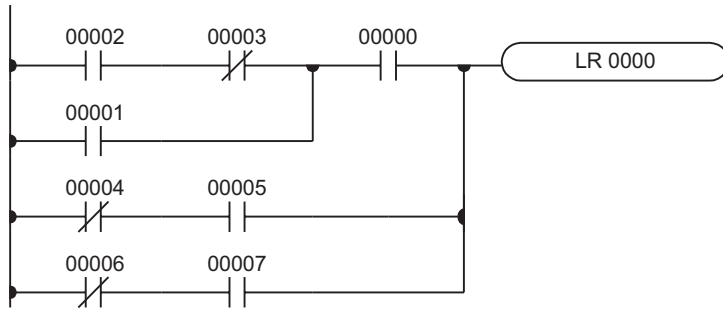
Следующая диаграмма требует команды OR LOAD, затем AND LOAD для кодирования верхних трех модуля, и далее два OR LOAD для завершения мнемкокода.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	LD	00001
00002	LD	00002
00003	AND NOT	00003
00004	OR LD	-
00005	AND LD	-
00006	LD NOT	00004
00007	AND	00005
00008	OR LD	-
00009	LD NOT	00006
00010	AND	00007
00011	OR LD	-
0012	OUT	LR 0000

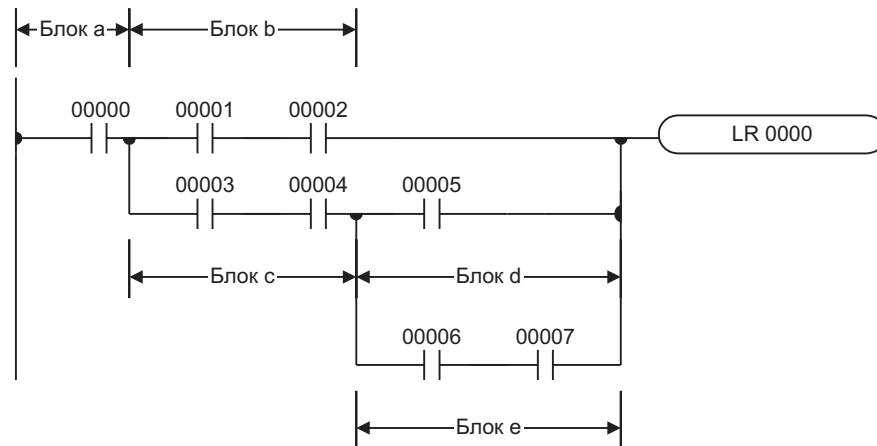
#### 4.4 Базовые релейно-контактные схемы

Хотя программа будет выполняться, как и написано, для устранения необходимости первого OR LOAD и AND LOAD данную диаграмму можно переписать, упростив программу и сэкономив память.



Адрес	Инструкция	Операнд
00000	LD	00002
00001	AND NOT	00003
00002	OR	00001
00003	AND	00000
00004	LD NOT	00004
00005	AND	00005
00006	OR LD	-
00007	LD NOT	00006
00008	AND	00007
00009	OR LD	-
00010	OUT	LR 0000

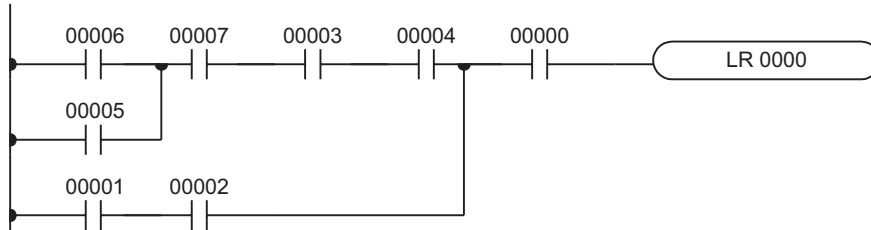
Следующая диаграмма требует 5 модулей, которые здесь кодируются по порядку перед использованием OR LOAD и AND LOAD для их объединения, начиная с последних двух модулей и двигаясь назад. OR LOAD по адресу 00008 объединяет модули d и e, следующая AND LOAD объединяет результат с результатом модуля с и т. д.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	LD	00001
00002	AND	00002
00003	LD	00003
00004	AND	00004
00005	LD	00005
00006	LD	00006

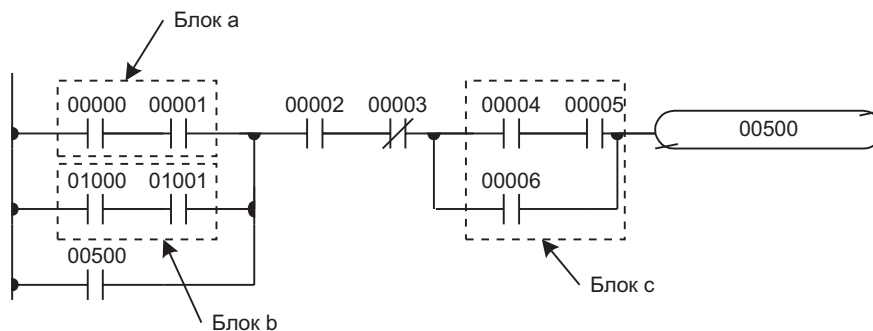
Адрес	Инструкция	Операнд
00007	AND	00007
00008	OR LD	-(блок d и e)
00009	AND LD	-(блок с с результатом предыдущей операции)
00010	OR LD	-(блок b с результатом предыдущей операции)
00011	AND LD	-(блок a с результатом предыдущей операции)
00012	OUT	LR 0000

Эту диаграмму можно переделать в следующий вид для упрощения структуры и кодирования программы и экономии памяти.

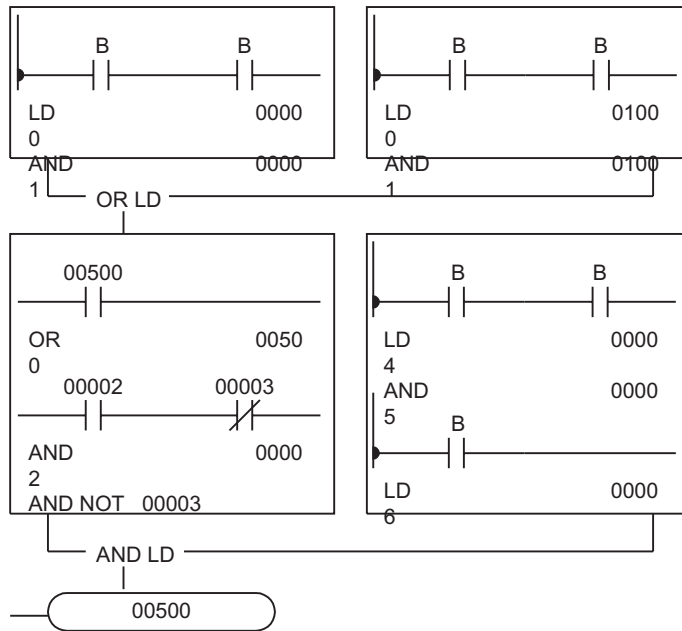


Адрес	Инструкция	Операнд
00000	LD	00006
00001	AND	00007
00002	OR	00005
00003	AND	00003
00004	AND	00004
00005	LD	00001
00006	AND	00002
00007	OR LD	-
00008	AND	00000
00009	OUT	LR 0000

Следующий и последний пример может сначала показаться слишком сложным, но кодирование можно произвести только двумя блоковыми командами. Диаграмма имеет следующий вид:



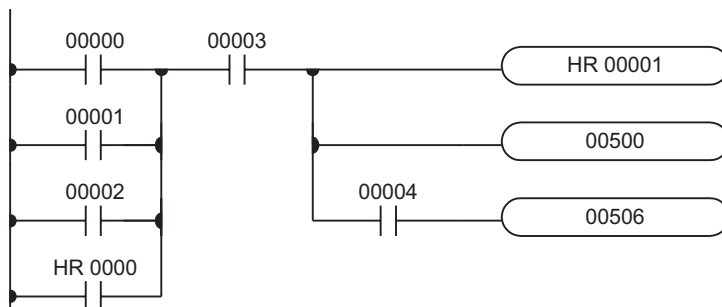
Первая блоковая команда служит для объединения модулей a и b, а вторая - для объединения модуля с нормально закрытым условием IR 00003. Оставшуюся часть диаграммы можно закодировать командами OR, AND и AND NOT. Далее показан алгоритм и результирующий код.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND	00001
00002	LD	01000
00003	AND	01001
00004	OR LD	-
00005	OR	00500
00006	AND	00002
00007	AND NOT	00003
00008	LD	00004
00009	AND	00005
00010	OR	00006
00011	AND LD	-
00012	OUT	00500

#### 4.4.7 Кодирование нескольких выходных команд

Если несколько выходных команд выполняются при одном условии исполнения, их нужно кодировать последовательно, вслед за последним условием командной линии. В следующем примере последняя строка команд содержит еще одно условие (ЛОГИЧЕСКОЕ И с IR 00004).



Адрес	Инструкция	Операнд
00000	LD	00000
00001	OR	00001

Адрес	Инструкция	Операнд
00002	OR	00002
00003	OR	HR 0000
00004	AND	00003
00005	OUT	HR 0001
00006	OUT	00500
00007	AND	00004
00008	OUT	00506

## 4.5 Консоль программирования

В данной и последующих главах описана Консоль программирования и операции, необходимые для подготовки ввода программы. Процедуры ввода программы в память см. 4.7.

Хотя Консоль программирования можно использовать для написания релейно-контактных программ, он, прежде всего, используется для поддержки операций ПО и весьма полезен для редактирования программ на месте и их обслуживание. Далее перечисляются функции программатора.

- 1, 2, 3,... 1. Индикация рабочих сообщений и результатов диагностических проверок.
2. Запись и чтение релейно-контактных программ, вставка и удаление команд, поиск параметров и команд, индикация состояния битов входов/выходов.
3. Индикация состояния входов/выходов, принудительная установка/сброс битов.
4. Консоль программирования можно подключать или отключать от ПК с поданным питанием.
5. Консоль программирования можно использовать с ПК серии С.
6. Поддерживает режим TERMINAL, который позволяет индцировать сообщения из 32 знаков, а также имеет функцию отображения клавиатуры. Подробности см. 5.25.6.

**Замечание** Консоль программирования поддерживает не все операции ПО, а только требующие редактирования и обслуживания на месте.

### 4.5.1 Клавиатура

Клавиатура программатора функционально разделяется по цвету клавиш на следующие 4 области:

#### **Белый: Цифровые клавиши**

Десять белых клавиш используются для ввода цифровых данных, таких как адреса программ, адреса параметров и значения операндов. Цифровые клавиши используются также в комбинации с функциональными клавишами (FUN) для ввода команд с функциональными кодами.

#### **Красный: Клавиша CLR (очистка)**

Клавиша CLR очищает дисплей и отменяет текущую операцию программатора. Она также используется, когда Вы вводите пароль в начале операций программирования. Любую операцию программатора можно отменить, нажав клавишу CLR, хотя возможно, что клавишу CLR придется нажать два или три раза для отмены операции и очистки дисплея.

#### **Желтый: Клавиши операции**

Желтые клавиши используются для записи и изменения программ. Подробные объяснения их функций приведены далее в данной главе.

**Серый: Клавиши команды и области данных**

За исключением клавиши SHIFT слева вверху, серые клавиши используются для ввода команд и задания префиксов областей, данных при вводе и изменении программы. Клавиша SHIFT аналогична клавише сдвига на печатной машинке и служит для изменения функции следующей нажатой клавиши. Нет необходимости удерживать клавишу SHIFT. Достаточно нажать ее один раз и затем нажимать следующую клавишу.

Серые клавиши, кроме SHIFT, имеют либо мнемония команды, либо сокращенное название области данных, которые написаны на них. Функции этих клавиш:



Нажимается перед функциональным кодом при вводе команды ее функциональным кодом.



Нажимается для ввода SFT (Команда сдвига регистра)



Вводится либо после функционального кода для назначения формата команды фронта 0\1 (@) или после релейно-контактной команды для назначения инверсного условия.



Нажимается для ввода AND (команда END) или используется с NOT для ввода AND NOT.



Нажимается для ввода OR (команда OR) или используется с NOT для ввода OR NOT.



Нажимается для ввода CNT (команда счетчика) или для обозначения номера ТС, который уже назначен как счетчик.



Нажимается для ввода LD (команда загрузки) или используется с NOT для ввода LD NOT. Нажимается также для индикации входного бита.



Нажимается для ввода OUT (команда вывода) или используется с NOT для ввода OUT NOT. Нажимается также для индикации выходного бита.



Нажимается для ввода TIM (команда таймера) или для обозначения номера ТС, который уже назначен как таймер.



Нажимается перед назначением адреса в области TR.



Нажимается перед назначением адреса в области LR.



Нажимается перед назначением адреса в области HR.



Нажимается перед назначением адреса в области AR.



Нажимается перед назначением адреса в области DM.



Нажимается перед назначением адреса в области EM.



Нажимается перед назначением косвенного адреса DM.



Нажимается перед назначением адреса слова.



Нажимается перед назначением операнда как константы.



Нажимается перед назначением адреса бита.



Нажимается перед функциональным кодом для блоковых команд, т.е. расположенных между скобками.

#### 4.5.2 Режимы работы ПК

На программаторе имеется переключатель для управления режимами ПК. Для выбора одного из трех режимов - RUN, MONITOR или PROGRAM - пользуйтесь этим переключателем. Режим, который Вы выбираете, будет определять работу ПК и процедуры, которые возможно проводить с программатора.

RUN - рабочий режим ПК. Когда переключатель установлен на RUN и вход START на модуле питания ЦПУ = 1, ЦПУ начнет выполнять программу, записанную в памяти программ. Хотя в режиме RUN возможен просмотр операций ПК, нельзя вводить и изменять параметры в области памяти.



Режим MONITOR позволяет наблюдать выполнение программы, состояние входов/выходов, изменять текущее значение или задание. В режиме MONITOR обработка входов/выходов производится аналогично режиму RUN. Режим MONITOR используется главным образом для проверки системы и окончательной отладки программы.

В Режиме PROGRAM программа не выполняется. Режим PROGRAM используется для написания и редактирования программ, очистки памяти и регистрации и изменения таблицы входов/выходов. В режиме PROGRAM имеется также специальная операция отладки, она позволяет проверку программу перед ее опробованием в системе.

**Внимание!** Не оставляйте Консоль программирования подключенной к ПК с помощью дополнительного кабеля, когда он находится в режиме RUN. Помехи, создаваемые дополнительным кабелем, могут попасть в ПК, тем самым воздействуя на работу программы и, следовательно, на систему.

### 4.5.3 Переключатель языка сообщений

Секция 3 на переключателе DIP ЦПУ определяет язык сообщений на программаторе (английский или японский). На заводе секция устанавливается в ON (английский язык).

## 4.6 Подготовка к работе

В данной главе описаны процедуры начала работы с программатором. Они включают ввод пароля, очистку памяти, очистку сообщений об ошибках и операции с таблицей входов/выходов. Операции с таблицей входов/выходов также необходимы и в других случаях, т.е. когда необходимо сделать изменения в модулях, используемых в конфигурации ПК.

**Внимание!** При включении ПК с подключенным программатором всегда смотрите, чтобы Консоль программирования находилась в режиме PROGRAM (если только не требуется другой режим для специальных целей). Если при включении ПК с подключенным программатором Консоль программирования находится в режиме RUN, будет выполняться программа, записанная в памяти программ, тем самым вызвав работу системы.

Перед началом первого ввода программы следует выполнить последовательность операций:

- 1, 2, 3,... 1. Вставьте ключ переключателя режимов в Консоль программирования.
2. Установите переключатель режимов на PROGRAM. (При работе в режиме PROGRAM ключ удалить нельзя).
3. Включите ПК.

**Замечание** Когда установлены модули входов/выходов, также включите эти модули. Если они не включены, Консоль программирования работать не будет.

4. Убедитесь, что на ЦПУ горит светодиод POWER и на экране программатора появилось следующее изображение. (Если горит или мигает светодиод ALM/ERR или появилось сообщение об ошибке, очистите сообщение).

< PROGRAM >  
PASSWORD!

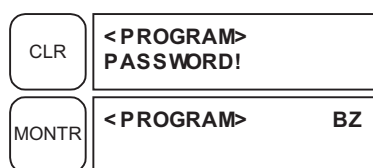
5. Введите пароль. Подробности см. 4.6.1
6. Очистите память. Если программу из памяти убирать не нужно, пропустите этот шаг. Подробности см. 4.6.3.

### 4.6.1 Ввод пароля

Для получения доступа к функциям программирования ПК Вы должны вначале ввести пароль. Пароль предотвращает несанкционированный доступ к программе.

ПК требует ввода пароля при включении ПК или, если ПК уже включен, после подключения программатора к ПК. Для получения разрешения на работу, после появления сообщения "PASSWORD" нажмите CLR и MONTR. Затем нажмите CLR для очистки дисплея.

Если Консоль программирования подключается к ПК, когда ПК уже включен, сначала будет индицироваться режим, в котором работал ПК перед подключением программатора. Перед вводом пароля убедитесь, что ПК находится в режиме PROGRAM. Когда пароль введен, ПК переключится в режим, установленный на переключателе режимов, тем самым вызывая начало отработки программы ПК, если установлен режим RUN и MONITOR. После ввода пароля режим можно переключить на RUN или MONITOR.



### 4.6.2 Зуммер

Сразу после ввода пароля или в любое время после изменения режима можно нажать SHIFT и 1 для включения и выключения зуммера, который звучит при нажатии клавиш программатора. Если в правом верхнем углу высвечивается BZ, зуммер активен. Если BZ не высвечивается BZ, зуммер не активен.

Зуммер также звучит каждый раз при появлении ошибки при работе ПК. На работу зуммера при ошибках эта установка не влияет.

### 4.6.3 Очистка памяти

С помощью программатора можно очистить всю или часть области UM (RAM или EEPROM) и области IR, HR, AR, DM, EM и TC. Если нет специальных указаний, операция очистки очищает все перечисленные области памяти. Область UM не очищается, если переключатель защиты записи находится в состоянии ON (секция 1 переключателя DIP на ЦПУ).

Перед первым программированием или введением новой программы все области памяти должны быть очищены. Перед очисткой проверьте, не загружена ли нужная программа. Если программа Вам нужна, очистите только те области памяти, которые Вам не нужно сохранять, и обязательно проведите процедуру проверки существующей программы перед ее использованием. Порядок проверки описан далее в данной главе. Методы дальнейшей отладки описаны в гл. 7. Для очистки всех областей памяти нажимайте CLR, пока не будут индицироваться все нули, а затем клавиши, указанные в верхней строке последовательности клавиш. Ответвления, показанные в последовательности, применяются только при выполнении частичной очистки клавиш, которая описана далее.

Память можно очищать только в режиме PROGRAM. В таблице показано, какие области памяти будут очищены в трех режимах очистки (полная очистка, частичная ошибка, очистка памяти).

Область памяти	Полная очистка	Частичная очистка	Очистка памяти
Слова входа/выхода	Очищается	Очищается	Очищается
Рабочие слова	Очищается	-	Очищается
HR, AR, TC, DM, фиксированные DM	Очищается	Очищается	Очищается

Область памяти	Полная очистка	Частичная очистка	Очистка памяти
Дополнительные DM	Очищается	-	Очищается
EM	Очищается	Очищается	Очищается
Комментарии к входам / выходам	Очищается	-	-
Релейно-контактная программа	Очищается	Очищается	Очищается
Информация распределения UM	Очищается	-	-

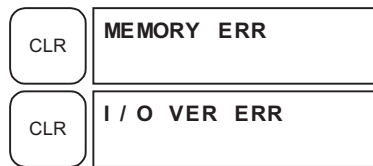
- Замечание**
1. Область протокола ошибок (DM 6 000..DM 6 030) не будет очищена, когда очищается область DM.
  2. При очистке области установочных параметров (DM 6600..DM 6655 в фиксированных DM), значения параметров будут возвращены к значениям по умолчанию.
  3. При выполнении операции полной очистки область релейно-контактных программ будет полностью выделена под релейно-контактные программы. (Дополнительные DM и области комментариев к входам/выходам будут установлены в 0 К слов). Также будут очищены все банки EM.

**Полная очистка**

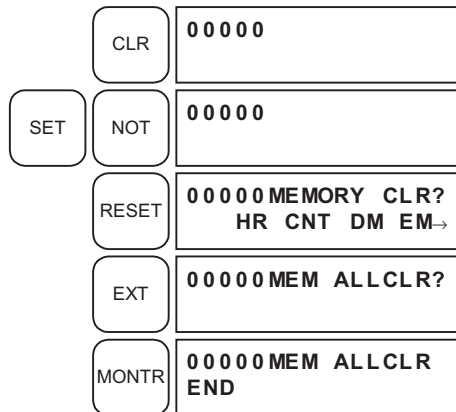
Далее показана последовательность нажатия клавиш для полной очистки.



Для полной очистки памяти служит следующая процедура.



Продолжайте нажимать клавишу CLR (один раз на каждое сообщение), пока на дисплее не появится 00000



**Частичная очистка**

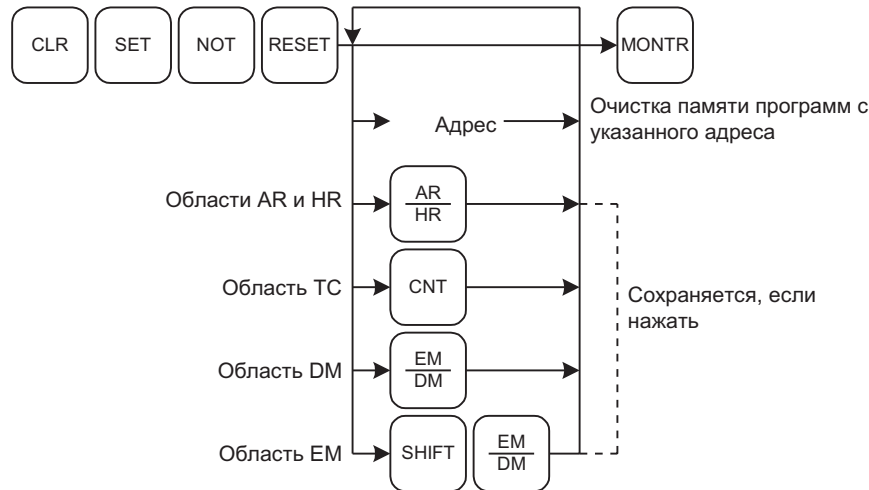
Имеется возможность сохранить параметры в указанных областях или часть релейно-контактной программы. Для сохранения данных в областях HR и AR, TC, DM и/или EM нажмите соответствующую клавишу после REC/RESET.

Клавиша HR служит для указания двух областей AR и HR, т.е. при сохранении области HR будет сохранена и область AR. Если нет указаний на сохранение, будут удалены обе области. CNT служит для всей области TC. Для задания области EM нажмите SHIFT+DM. На дисплее будут индцироваться области, которые будут очищены.

Имеется возможность сохранить некоторые банки EM и очистить другие. См. объяснения под заголовком “Очистка выбранных банков EM”.

Также возможно сохранение части релейно-контактной программы от начала до указанного адреса. После задания областей параметров, подлежащих сохранению, укажите первый адрес памяти программ, подлежащий очистке. Например, введите 00123 чтобы оставить в сохранности адреса 00000-00122, но очистить память с адреса 00123 до конца области памяти программ.

Для частичной очистки памяти служит следующая последовательность клавиш:



Для того чтобы оставить область TC неочищенной и сохранить адреса памяти программ 0000..00122, введите следующее:

CLR	00000
SET	00000
NOT	00000
RESET	00000MEMORY CLR? HR CNT DM EM
CNT	00000MEMORY CLR? HR DM EM
B 1 C 2 D 3	00123MEMORY CLR? HR DM EM
MONTR	00000MEMORY CLR END HR DM EM

#### Очистка выбранных банков EM

Когда выполняется операция частичной очистки памяти можно задать банки для очистки, а не очищать всю память EM. В следующем примере для очистки выбраны банки EM 0 и 2.

На программаторе будут следующие изображения:

00000
00000
00000
00000MEMORY CLR? HR CNT DM EM
00000 EM CLR ? 012
00000 EM CLR ? 0 2
00000 EM CLR END 0 2

**Очистка памяти**

Операция очистки памяти очищает все области памяти, кроме комментариев к входам/выходам и информацию о распределении UM.

На программаторе будут следующие изображения:

CLR	00000
SET	00000
NOT	00000
RESET	00000MEMORY CLR? HR CNT DM EM
MONTR	00000MEMORY CLR END

**Замечание** Когда переключатель защиты от записи (секция 1 переключателя DIP на ЦПУ) установлен в ON, область UM (от DM 6144 до конца релейно-контактной программы) не будет очищаться. Другие области, такие как HR, AR, CNT, и DM 0000..DM 6143 будут очищены.

**4.6.4 Регистрация таблицы входов/выходов**

Операция регистрации таблицы входов/выходов записывает типы модулей входов/выходов, которыми управляет ПК, и расположение модулей входов/выходов на панели. Она также очищает все биты входов/выходов.

Регистрировать таблицу входов/выходов необязательно у C200HX/HG/HE. Когда таблица входов/выходов не зарегистрирована, при включении питания ПК будет работать согласно модулям входов/выходов. Ошибка проверки/настройки входов/выходов не появляется.

Регистрировать таблицу входов/выходов обязательно, если модули входов/выходов меняются, в противном случае в начале операций программирования появляется сообщение об ошибке I/O VER ERR или I/O SET ERROR.

Регистрацию таблицы входов/выходов можно проводить только в режиме PROGRAM, когда переключатель защиты от записи (секция 1 переключателя DIP) установлен в OFF. (OFF = запись).

**Последовательность нажатия клавиш**

Начальная регистрация таблицы входов/выходов

	CLR	00000
	FUN	00000 FUN ( ?? )
SHIFT	CH/*DM	000001 OTBL ? ?- ?U=
	CHG	000001 OTBL WRI T ????
9 7 B 1 D 3		000001 OTBL WRI T 9713
	WRITE	000001 OTBL WRI T OK

**4.6.5 Очистка сообщений об ошибках**

После регистрации таблицы входов/выходов следует очистить все сообщения об ошибках, записанных в памяти. Предполагается, что причины ошибок уже устранены. Если при попытке убрать сообщение звенит зуммер, устраните причину ошибки и потом сотрите сообщение (см. гл. 9).

Для индикации записанных сообщений об ошибках нажмите CLR, FUN и затем MONTR. Появится первое сообщение. Повторное нажатие MONTR удалит текущее сообщение и вызовет следующее. Продолжайте нажимать MONTR до тех пор, пока не уберутся все сообщения.

Хотя сообщения о фатальных ошибках высвечиваются во всех режимах, убрать их можно только в режиме PROGRAM.

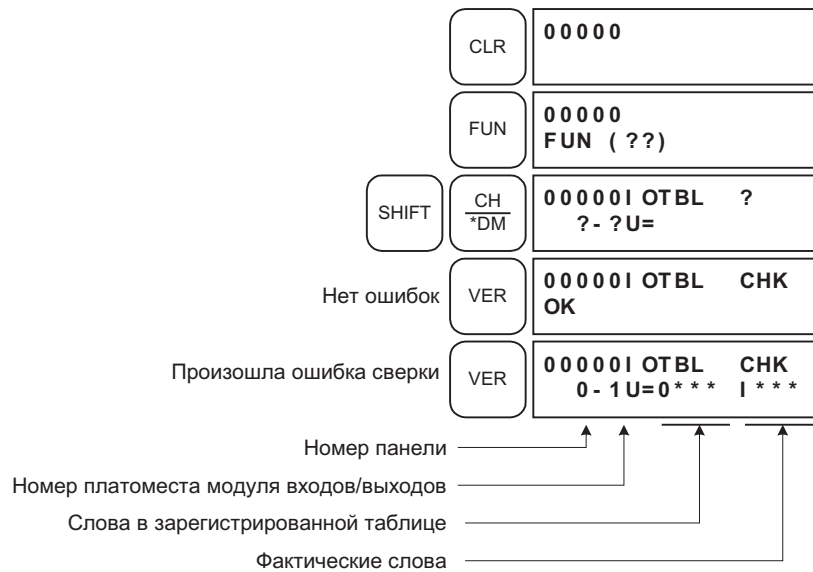
**Последовательность нажатий клавиш****4.6.6 Сверка таблицы входов/выходов**

Сверка таблицы входов/выходов служит для сверки таблицы входов/выходов, зарегистрированной в памяти, соответствует ли она фактической последовательности модулей входов/выходов. Первое несоответствие будет индцироваться, как будет показано далее. Каждое последующее нажатие VER высвечивает следующее несоответствие.

**Замечание** Данная операция может выполняться только когда таблица входов/выходов зарегистрирована.

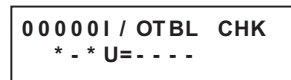
**Последовательность действий**

**Пример**

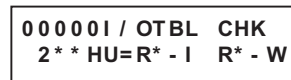


**Значение индикации**

Следующее изображение показывает, что С500, С1000Н, или С2000Н и С200Н, С200НС, или С200НХ/НГ/НЕ имеют одинаковые номера модулей на ведомой панели удаленных входов/выходов.



Следующее изображение показывает дублирование номеров модулей оптических входов/выходов.

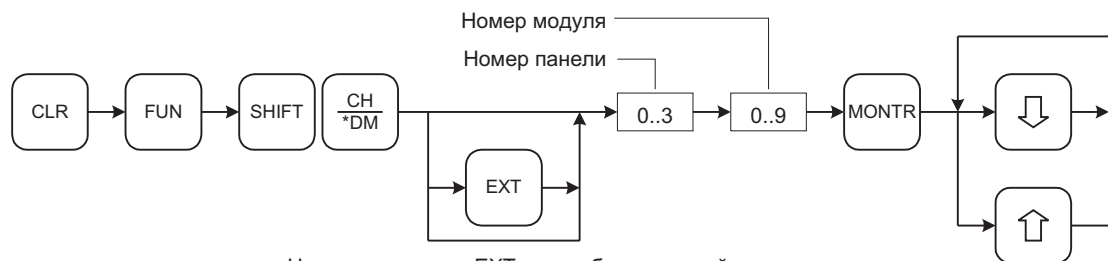


Указывает на дублирование

**4.6.7 Чтение таблицы входов/выходов**

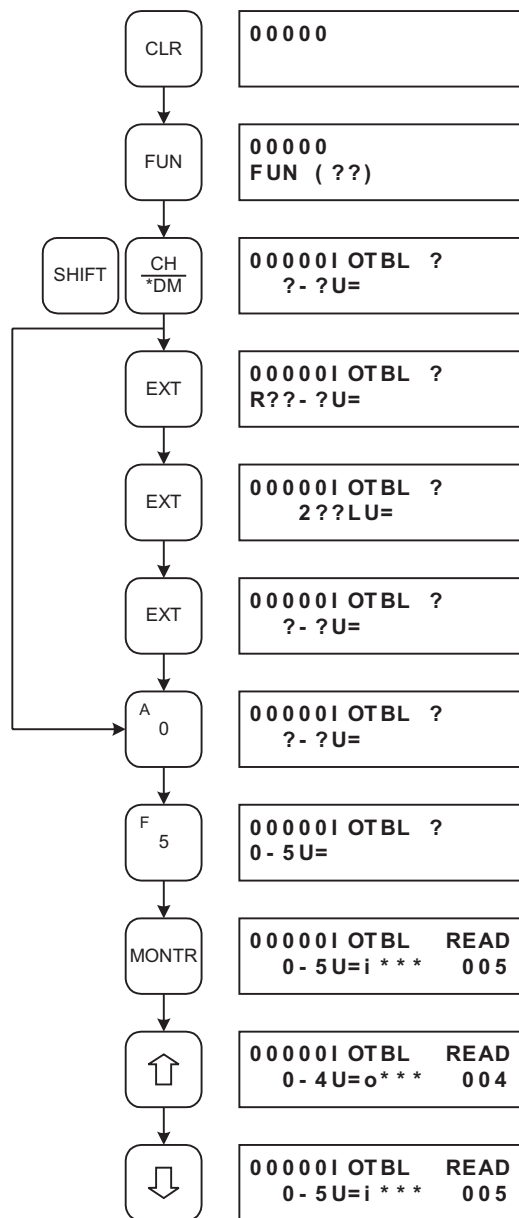
Операция чтения таблицы входов/выходов служит для доступа к таблице входов/выходов, зарегистрированной в памяти ЦПУ. Данная операция может выполняться в любом режиме ПК.

**Последовательность нажатия клавиш**



Нажмите клавишу EXT для выбора панелей удаленных входов/выходов или блок оптических входов/выходов

Пример

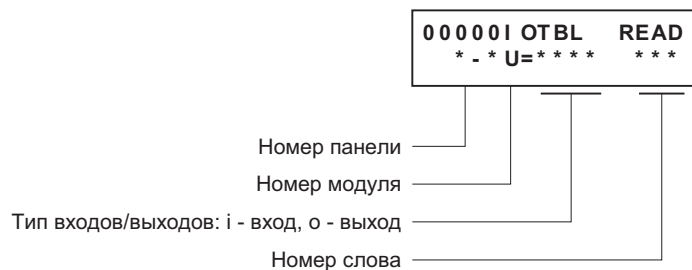


Значение сообщений

Обозначения модулей входов/выходов на сообщениях

Число точек	Модули входов	Модули выходов
8	I (*) **	O ***
16	II **	OO **

Модули входов/выходов





000001 OTBL READ  
\* . \* U=\*\*\*\*\*

INT0..INT1: модули прерываний 1 или 2

000001 OTBL READ  
\* . \* U=\$\*\*\*

Указывает на блок специальных входов/выходов

Номер модуля (0..F)

Тип модуля: С - счетчик, А - модуль позиционирования, А - другие

Пробел: только блок 1, W - только блок 2

000001 OTBL READ  
\* . \* U=RMT\*

Номер ведущего удаленной системы ввода/вывода (0..1)

000001 OTBL READ  
R\* . \* U=\*\*\*\*\*

Указывает панель удаленных входов/выходов

Номер ведущего модуля (0..1)

Номер ведомого модуля (0..4)

Номер модуля (0..9)

Тип входов/выходов: i - вход, o - выход

Номер слова

000001 OTBL READ  
\* . \* U=#\*\*\*

Указывает на модули группы 2

Номер модуля (0..F)

Тип входов/выходов: i - вход, o - выход

2: 2 слова (32 точки); 4: 4 слова (64 точки)

000001 OTBL READ  
2\*\* HU=R\* - \*

Номер слова (200..231)

Номер ведущего блока (0..1)

Слово (H: старшие 8 бит; L: младшие 8 бит)

Тип входов/выходов: i - вход, o - выход

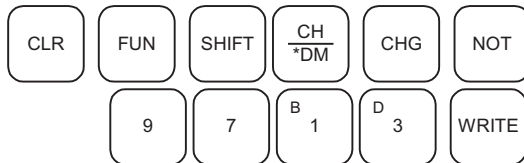
#### 4.6.8 Очистка таблицы входов/выходов

Операция очистки таблицы входов/выходов используется для удаления содержимого таблицы входов/выходов, зарегистрированной в памяти ЦПУ. После выполнения операции очистки таблицы входов/выходов ПК будет настроен на операцию, основанную на установленных модулях входов/выходов.

Операция очистки таблицы входов/выходов сбросит все установленные специальные модули входов/выходов и модули связи. Не проводите операцию очистки таблицы входов/выходов, когда работают модули связи HOST LINK, Модуль PC LINK, Модуль удаленного ведущего входов/выходов, Модуль высокоскоростного счетчика, Модуль позиционирования или другие специальных модули входов/выходов.

**Замечание** Данная операция может выполняться только в режиме PROGRAM, когда переключатель защиты от записи (секция 1 переключателя DIP) установлен в OFF. (OFF = запись).

##### Последовательность нажатия клавиш



##### Пример

CLR	00000
FUN	00000 FUN ( ?? )
SHIFT	CH *DM 000001 OTBL ? - ?U=
CHG	000001 OTBL    WRI T ????
NOT	000001 OTB    CANC ????
9    7    B 1    D 3	000001 OTB    CANC 9713
WRITE	000001 OTBL    CANC OK

#### 4.6.9 Передача таблицы связи SYSMAC NET

Операция передача таблицы связи SYSMAC NET передает копию таблицы связи SYSMAC NET в область памяти программ UM. Это позволяет записать совместно в EPROM программу пользователя и таблицу связи SYSMAC NET. Таблица связи SYSMAC NET должна быть создана с помощью ПО и передана в ПК перед копированием в память программ.

После загрузки из ПО в ПК таблица параметров связи хранится в ОЗУ, так что она будет потеряна при неисправности аккумулятора. Чтобы этого не случилось, мы рекомендуем Вам перевести программу (с таблицей параметров связи) в EPROM или сохранить программу на EEPROM кассеты памяти.

**Замечание** Когда подается питание на ПК, в котором имеется копия таблицы связи SYSMAC NET в памяти программ, таблица связи SYSMAC NET в ЦПУ будет переписана. Изменения, сделанные в таблице связи SYSMAC NET, не влияют на ее копию в памяти программ; для внесения изменений в памяти программ передача таблицы связи SYSMAC NET должна быть повторена.

Операция передача таблицы связи SYSMAC NET не будет действовать в следующих случаях:

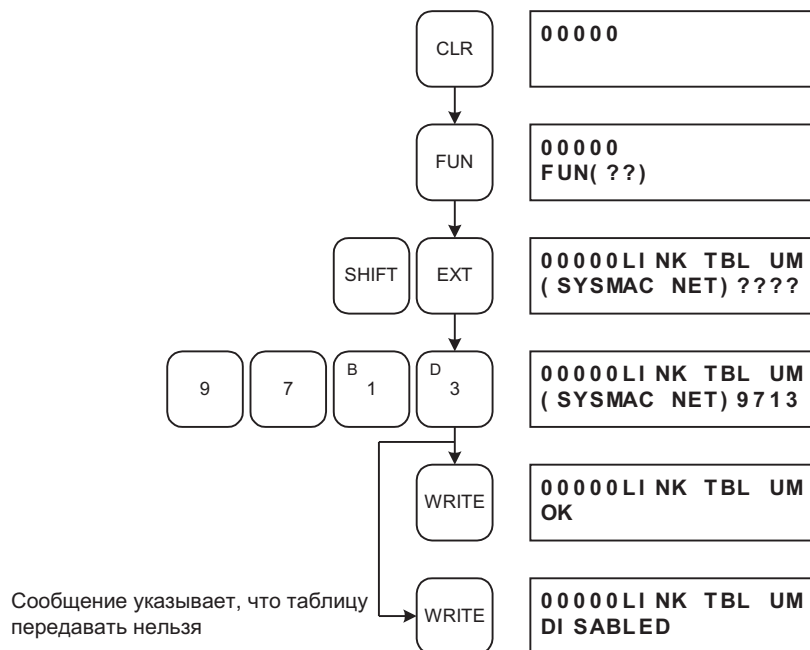
- Модуль памяти - не RAM или EEPROM, или переключатель защиты от записи не разрешает запись.
- Нет команды END.
- Содержимое памяти программ превышает 14.7 К слов. Объем памяти программ сократится, когда память выделяется на дополнительные DM или на комментарии к входам/выходам. Для сохранения таблицы связи требуется около 0.5 К слов после команды END.

Операция передача таблицы связи SYSMAC NET возможно только в режиме PROGRAM.

**Последовательность нажатия клавиш**



**Пример**



**4.7 Ввод, изменение и проверка программы**

Когда программа записана в мнемокоде, ее можно ввести в ПК прямо с программатора. Мнемокоды заносятся по адресам памяти программ с программатора. Проверка программы включает проверку на синтаксис, т.е. написана ли она в соответствии с правилами синтаксиса. Когда синтаксические ошибки исправлены, можно начать пробное исполнение и, наконец, можно проводить коррекцию под фактические условия работы.

В данном разделе объясняются операции, требуемые для ввода программы, изменения уже введенной программы и процедуры для чтения текущего времени цикла.

Перед началом ввода программы убедитесь, что программа не загружена. Если загружена программа, которая Вам не нужна, сначала очистите память, и только затем вводите новую программу. Если предыдущая программа Вам нужна, проверьте ее с помощью процедуры проверки и в случае необходимости исправьте. Дополнительные методы отладки изложены в гл. 7.

### 4.7.1 Установка и чтение из адреса памяти программ

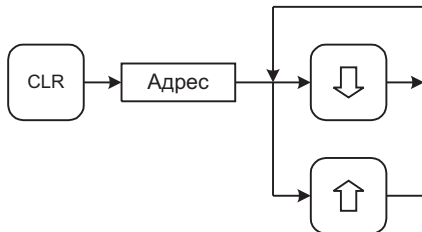
При первом вводе программы она, как правило, пишется в память программ, начиная с адреса 00000. Поскольку данный адрес появляется при очистке дисплея, нет необходимости задавать его.

При вводе программы в адрес, отличный от 00000, или при чтении или изменении уже существующей программы, необходимо задавать требуемый адрес. Для задания адреса нажмите CLR и затем введите требуемый адрес. Ведущие нули вводить необязательно, т.е. задавая адрес 00053 достаточно ввести 53. Содержание указанного адреса появится только после нажатия клавиши КУРСОР ВНИЗ.

После нажатия клавиши КУРСОР ВНИЗ для высвечивания содержимого по заданному адресу можно использовать клавиши КУРСОР ВНИЗ и КУРСОР ВВЕРХ для просмотра программы. При нажатии на эти клавиши будет высвечиваться предыдущий или последующий адрес программы.

При чтении программы в режиме RUN или MONITOR будет индицироваться состояние бита OFF(0) или ON (1).

#### Последовательность нажатия клавиш



#### Пример

			CLR	00000
C 2	A 0	A 0		00200
		↓	00200 READ AND	ON 00001
		↓	00201 READ AND	ON 00001
		↓	00202 READ TI M	OF 000
		↓	00202 TI M	#0123
		↓	00203 READ LD	ON 00100

Если в памяти программ уже находится мнемокод, приведенный в таблице, операция чтения даст следующее.

### 4.7.2 Ввод и редактирование программ

Программы можно вводить и редактировать только в режиме PROGRAM, когда переключатель защиты от записи (секция 1 переключателя DIP) установлен в OFF. (OFF = запись).

Одна и та же процедура используется как для ввода программы в первый раз, так и для изменения уже существующей программы. В обоих случаях содержимое памяти программ переписывается, т.е. если предыдущей программы нет, будут переписаны команды NOP, которые были записаны по всем адресам.

Для ввода программы последовательно вводите мнемокод, полученный с помощью релейно-контактной схемы, убедившись, что ввод производится в правильный адрес. Если высвечивается правильный адрес, введите первое слово команды и нажмите WRITE. Далее введите требуемые операнды, нажимая после каждого WRITE, т.е. WRITE нажимается после каждой строки мнемокода. По нажатию WRITE в конце каждой строки набранная команда или операнд вводится и появляется следующее изображение. Если команда требует двух и более слов, на дисплее будет индицироваться требуемый операнд и его значение по умолчанию. Если команда требует только одно слово, появится следующий адрес. Продолжайте вводить все строки мнемокода, пока не будет введена вся программа.

При вводе численных значений операндов нет необходимости вводить ведущие нули. Ведущие нули требуются только при вводе функциональных кодов (см. далее). При вводе операндов следите, чтобы перед всеми областями данных (за исключением IR и SR) были нажаты клавиша соответствующей области, а при задании константы клавиша CONT/#. CONT/# не требуется для задания (SV) счетчика или таймера (см. далее). Область AR задается нажатием SHIFT и затем HR. Номер TC в качестве битового операнда (т.е. флага завершения) задаются нажатием TIM или CNT перед адресом, в зависимости от того, определяет TC счетчик или таймер. Для задания косвенной адресации DM нажмите CH/\* перед адресом (для задания косвенного адреса DM нет необходимости нажимать DM).

### Ввод задания (SV) для счетчиков и таймеров

Задание для счетчиков и таймеров, как правило, вводится как константа, хотя возможен ввод адреса слова, в котором содержится задание. При вводе задания как константы CONT/# не требуется; просто введите число и нажмите WRITE. Для задания слова нажмите CLR и затем введите адрес, как будет показано далее.

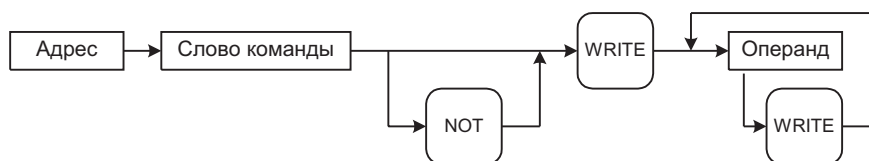
### Обозначение команд

Большинство базовых команд вводятся с помощью специальных клавиш программатора. Все остальные команды вводятся с помощью функциональных кодов. Эти коды всегда записываются после мнемоники команды. Если нет функционального кода, значит, для этой команды есть клавиша на программаторе. Для задания формы команды "фронт 0/1", после функционального кода нажмите NOT.

Для ввода команды с помощью функционального кода, задайте адрес, нажмите FUN, введите функциональный код, включая ведущие нули, нажмите NOT, если требуется форма команды "фронт 0/1", введите битовые операнды или определители, требуемые командой, и затем WRITE.

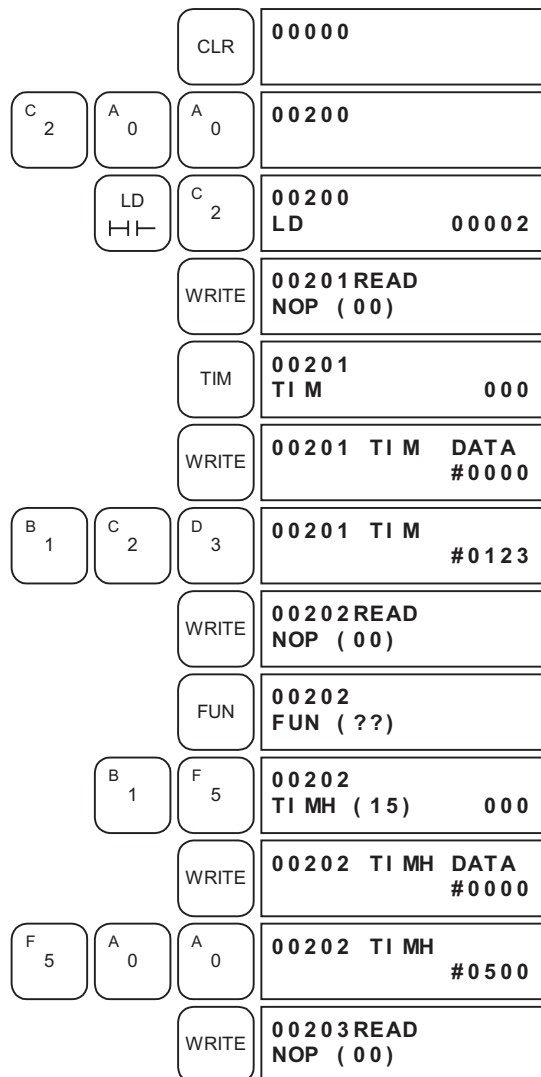
**Внимание!** Вводите функциональные коды внимательно и следите, чтобы при необходимости нажимался SHIFT.

### Последовательность нажатия клавиш



### Пример

Следующую программу можно ввести следующим порядком нажатия клавиш. Индикация будет иметь следующий вид.



Адрес	Инструкция	Операнд
00200	LD	00002
00201	TIM	000
		00123
00003	TIMH	000
		0500

**Сообщения об ошибках**

При вводе программы могут появиться следующие сообщения об ошибках. Исправьте ошибку, как указано, и продолжите операцию ввода. Звездочка на индикации будет заменяться числовыми значениями, как правило, адресом.

Сообщение	Значение сообщения и действия при сообщении
****REPL ROM	Была попытка записи в защищенную от записи память RAM или EEPROM. Установите переключатель защиты от записи в положение OFF (секция 1 переключателя DIP на ЦПУ)
****PROG OVER	Команда в последнем адресе памяти не NOP. Сотрите ненужные команды в хвосте программы
****ADDR OVER	Задан адрес, превышающий наибольший в памяти программ UM. Введите меньший адрес.
****SETDATA ERR	Был ввод данных в некорректном формате или за пределами дозволенных границ, например, вводилось 16-ричное значение вместо двоично-десятичного. Снова введите данные. Данная ошибка вызовет сообщение FALS 00.

Сообщение	Значение сообщения и действия при сообщении
****I/O NO. ERR	Был задан адрес, превышающий наибольший адрес зоны данных, т.е. адрес слишком велик. Изучите требования команды и снова введите адрес

### 4.7.3 Проверка программы

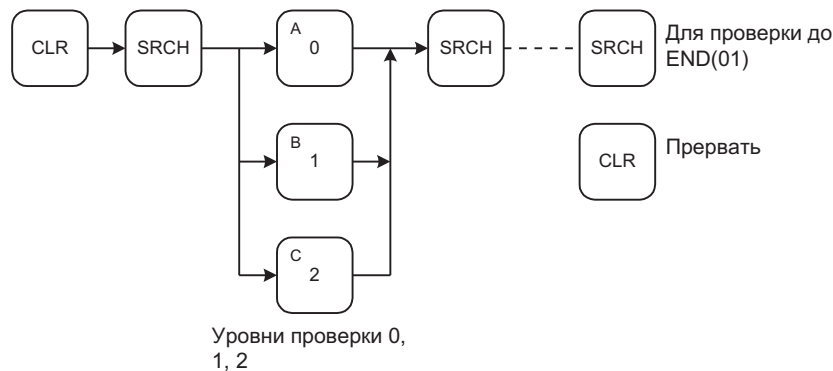
После ввода программы следует провести проверку синтаксиса, не нарушены ли правила программирования. Проверку следует проводить и после изменения программы, когда это могло привести к ошибке синтаксиса.

Для проверки программы сделайте операции, описанные далее. Числа указывают требуемый уровень проверки (см. далее). Когда уровень проверки введен, Проверка начнется. Если обнаружена ошибка, проверка остановится и на дисплее появится сообщение об ошибке. Для продолжения проверки нажмите SRCH. Если ошибок не обнаружено, программа будет проверяться до первой команды END. Для указанного после таблицы примера после проверки 64 команд на дисплее появится рисунок #1.

Для отмены проверки можно нажать CLR, и, как показано в примере, на дисплее появится рисунок #2. Когда будет достигнута первая команда END, появится рисунок #3.

Проверка синтаксиса можно проводить только в режиме PROGRAM.

#### Последовательность нажатия клавиш



#### Задание уровня и сообщений об ошибках

Имеется 3 уровня проверки программ. Перед проверкой должен быть задан необходимый уровень. В таблице приведены тип ошибок, сообщения и объяснения всех ошибок синтаксиса.

- Уровень контроля 0 проверяет на ошибки типа A, B, C.
- Уровень контроля 1 проверяет на ошибки типа A, B.
- Уровень контроля 2 проверяет на ошибки только типа A.

Будет также индексироваться адрес, по которому встретилась ошибка.

Многие из ошибок даны для команд, которые еще не описаны. Подробности о них см. 4.8 или 5.

Тип	Сообщение	Значение сообщения и порядок действий
Тип A	?????	Программа заперчена, появился несуществующий код. Введите программу снова.
	NO END INSTR	В программе отсутствует команда END. Запишите END в конце программы
	CIRCUIT ERR	Число логических модулей и команд логических модулей не совпадает, т.е. LD или LD NOT использованы для начала логического модуля, результаты которого не используются никакими другими командами, либо применена команда, для которой нет требуемого числа логических модулей. Проверьте программу.
	LOCKN ERR	Команда в неправильном месте программы. Изучите, как пользоваться командой и скорректируйте программу.

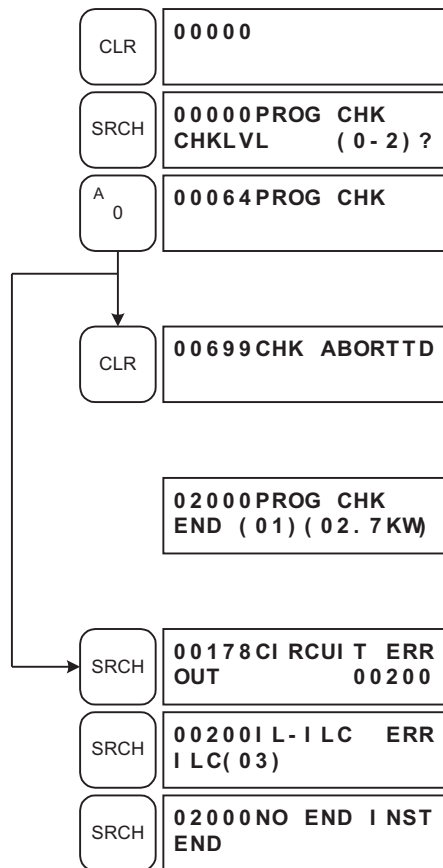
#### 4.7 Ввод, изменение и проверка программы

Тип	Сообщение	Значение сообщения и порядок действий
	DUPL	Дважды использован один и тот же номер перехода или подпрограммы. Скорректируйте программу, чтобы один номер использовался только для одной из них. (Переход JUMP 00 можно использовать сколько угодно раз).
	SBN UNDEFD	Команда SBS запрограммирована для несуществующей подпрограммы. Скорректируйте номер или запрограммируйте подпрограмму.
	JME UNDEFD	Команда JME отсутствует для команды JMP. Скорректируйте номер перехода или вставьте положенную команду JME.
	OPERAND ERR	Операнд команды, лежит вне допустимой зоны. Измените значение операнда, чтобы он лежал в допустимой зоне
	STEP ERR	Некорректно использованы STEP с номером секции и STEP без номера секции. Изучите работу команды. Скорректируйте программу.
Тип B	IL-ILC ERR	IL и ILC не используются попарно. Скорректируйте программу, чтобы у каждой IL был свой ILC. Хотя это сообщение об ошибке появится, если более одной IL используется с одной ILC, программа выполнится как написано. Перед отработкой убедитесь, что Ваша программа написана в соответствии с замыслом.
	JMP-JME ERR	JMP и JME не используются в паре. Перед отработкой убедитесь, что Ваша программа написана в соответствии с замыслом.
	SBN-RET ERR	Если индицируемый адрес - адрес SBN. 2 разные подпрограммы определены одним именем. Измените один из номеров или удалите одну из подпрограмм. Если индицируемый адрес - адрес RET. данная команда использована неправильно. Изучите работу команды RET и скорректируйте программу.
Тип C	JMP UNDEFD	JME(005) был использован с JMP (004) не с таким номером. Добавьте JMP (004) с таким же номером или удалите JME (005).
	SBS UNDEFD	Существует подпрограмма, которая не вызывается SBS(091). Запрограммируйте вызов подпрограммы в нужном месте или удалите подпрограмму, которая не используется.
	COIL DUPL	Один и тот же бит управляется (т.е. включается и/или выключается) более чем одной командой. (напр. OUT, OUT NOT, DIFU. DIFD. KEEP. SFT ). Хотя это разрешено для некоторых команд, просмотрите работу команды и или убедитесь, что программа написана правильно, или перепишите программу, чтобы каждый бит управлялся одной командой.

#### Пример

В следующем примере показаны изображения на дисплее, которые могут появиться при проверке программ.



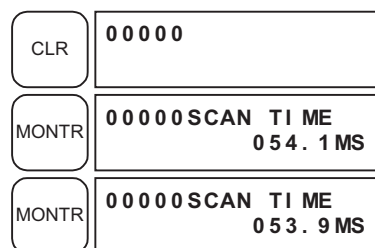


#### 4.7.4 Индикация времени цикла

Когда в программе устранены синтаксические ошибки следует проверить время цикла. Это возможно только в режиме RUN и MONITOR во время исполнения программы. Подробности о времени цикла см. гл. 6.

Для индикации среднего времени цикла нажмите CLR и MONTR. Время, высвечиваемое при этой операции, является средним временем цикла. Отличия в индицируемых значениях зависят от условий исполнения при нажатии MONTR.

##### Пример



#### 4.7.5 Поиск в программе

В программе можно производить поиск любой указанной команды или адреса области данных, используемого в команде. Поиск можно проводить с любого индицируемого адреса или от очищенного дисплея.

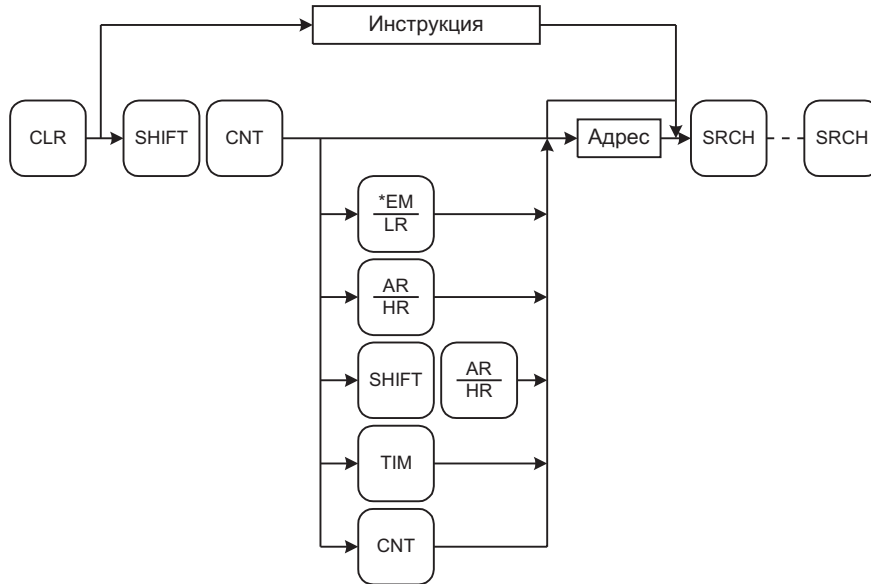
Для задания адреса бита нажмите SHIFT, CONT/#, далее адрес, включая имя области, и нажмите SRCH. Для задания команды введите команду (как и при вводе программы) и нажмите SRCH. Когда команда или адрес встретилась, последующие поиски той же команды или адреса производятся нажатием SRCH. Когда идет поиск, высвечивается SRCH'G.

## 4.7 Ввод, изменение и проверка программы

Когда при поиске операции, состоящей из нескольких слов, индицируется первое слово, для индикации следующих слов нажмите клавишу КУРСОР ВНИЗ.

Когда память программ читается в режиме RUN или MONITOR, будет высвечиваться состояние индицируемого бита.

### Последовательность нажатия клавиш



### Пример: Поиск команды

CLR	00000		
LD	00000 LD 00000		
SRCH	00200SRCH LD 00000		
SRCH	00202SRCH LD 00000		
SRCH	02000SRCH END (01)(02.7KW)		
CLR	00000		
B 1	A 0	A 0	00100
TIM	B 1		00100 TIM 001
SRCH			00203SRCH TIM 001
↓			00203 TIM DATA #0123

Пример: Поиск бита

	CLR	00000
SHIFT	CONT #	F 5
		00000CNT CONT 00005
	SRCH	00200CONT SRCH LD 00005
	SRCH	00203CONT SRCH AND 00005
	SRCH	02000 END (01)(02.7KW)

4.7.6 Вставка и удаление команд

В режиме PROGRAM любую команду, которая индицируется, можно удалить или вставить любую другую команду перед ней. Данные операции возможны только в режиме PROGRAM, когда переключатель защиты от записи (секция 1 переключателя DIP) установлен в OFF (OFF = запись).

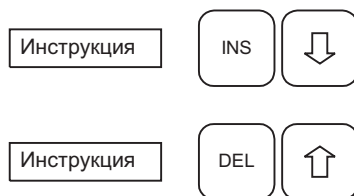
Для ввода команды высветите команду, перед которой Вы хотите вставить новую команду, введите слово команды (таким же образом, как и при первоначальном вводе программы) и нажмите INS и КУРСОР ВНИЗ. Если команда требует дополнительных слов, введите их таким же образом, как и при первоначальном вводе программы.

Для удаления команды высветите команду, которую Вы хотите удалить, нажмите DEL и КУРСОР ВНИЗ. Все слова указанной команды будут удалены.

**Внимание!** Случайно не удалите нужные команды. Их можно восстановить только повторным вводом.

Последовательность нажатия клавиш

Определите позицию в программе, затем введите

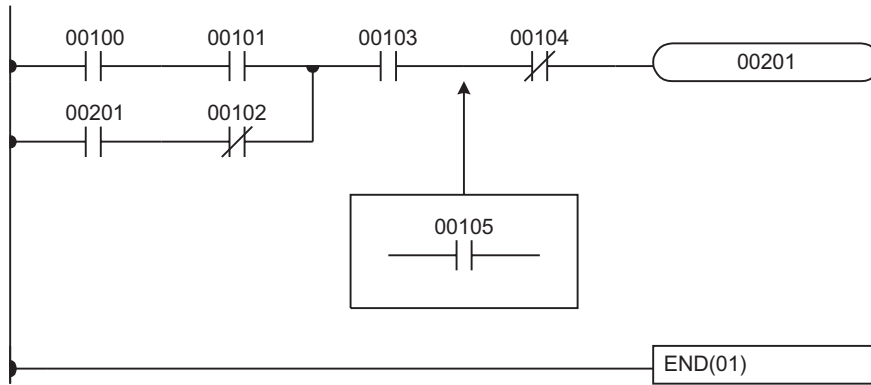


Когда вставляется или удаляется команда, все адреса в памяти программ настраиваются автоматически, чтобы не было пропущенных адресов или команд без адреса.

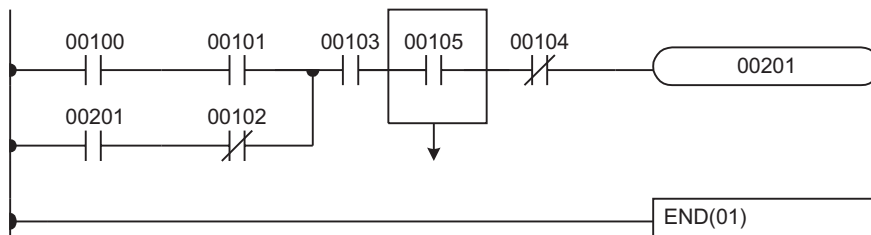
Пример

Следующий мнемокод показывает изменения, которые получаются в программе после проведения процедуры ввода/удаления, как показано ниже.

Перед вставкой



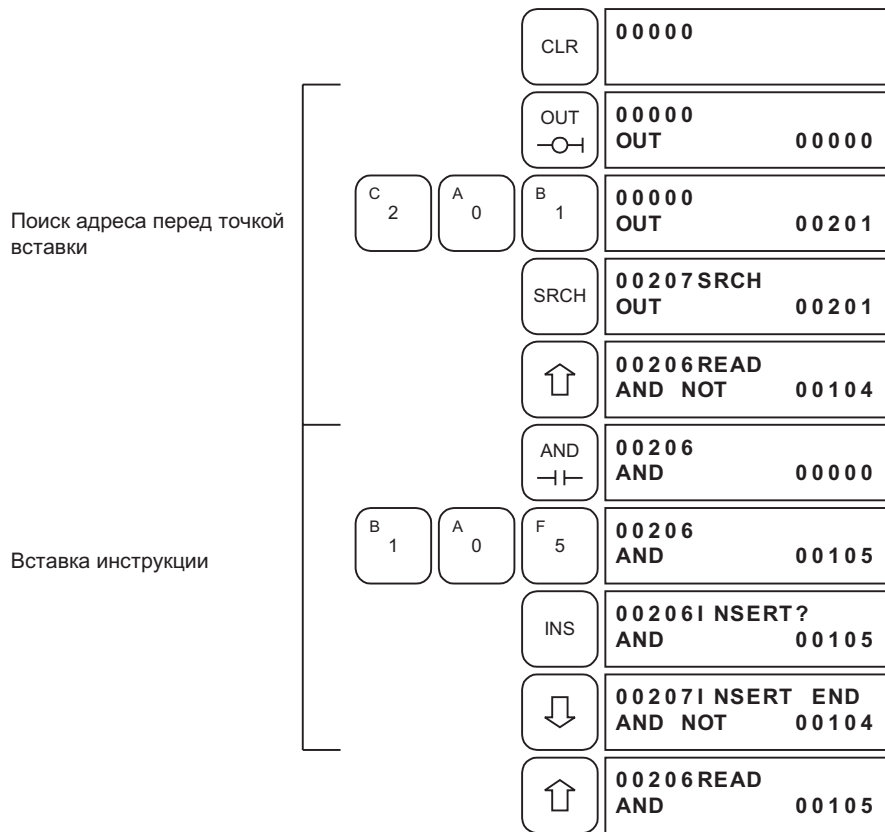
Перед удалением



Адрес	Инструкция	Операнд
00000	LD	00100
00001	AND	00101
00002	LD	00201
00003	AND NOT	00102
00004	OR LD	
00005	AND	00103
00006	AND NOT	00104
00007	OUT	00201
00008	END	

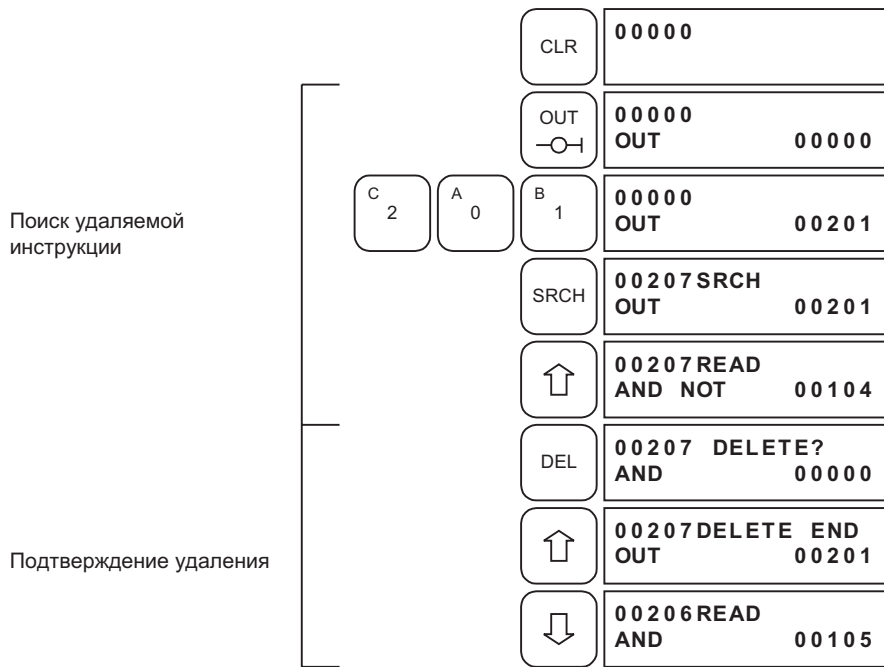
Следующая последовательность нажатия клавиш и индикации показывает, как достигаются изменения.

Ввод команды



Адрес	Инструкция	Операнд
00000	LD	00100
00001	AND	00101
00002	LD	00201
00003	AND NOT	00102
00004	OR LD	
00005	AND	00103
00006	AND	00105
00007	AND NOT	00104
00008	OUT	00201
00009	END	

Удаление команды

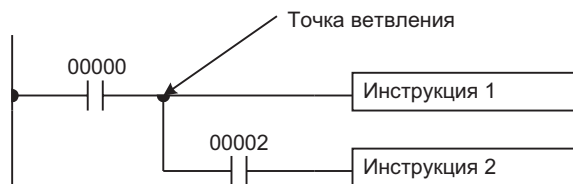


Адрес	Инструкция	Операнд
00000	LD	00100
00001	AND	00101
00002	LD	00201
00003	AND NOT	00102
00004	OR LD	
00005	AND	00103
00006	AND	00105
00007	OUT	00201
00008	END	

4.7.7 Разветвляющиеся командные линии

Когда командная линия разветвляется на 2 и более линий, иногда необходимо применить либо операцию INTERLOCK, либо биты TR для сохранения условия исполнения, которое существовало в точке ветвления. Это требуется оттого, что перед возвратом в точку ветвления для выполнения командной линии ветвления выполняются командные линии к “правосторонним” командам. Если существуют условия после точки ветвления, в это время результаты могут измениться, что сделает правильный результат невозможным. Следующая диаграмма иллюстрирует это. В обеих диаграммах команда 1 выполняется перед возвращением в точку ветвления и далее ветвление на команду 2.

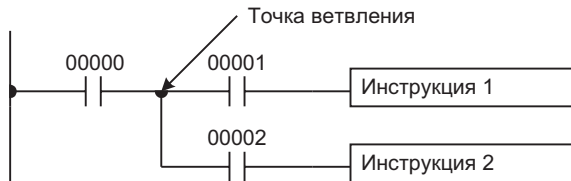
Корректная операция



Адрес	Инструкция	Операнд
00000	LD	00000

Адрес	Инструкция	Операнд
00001	Инструкция 1	
00002	AND	00002
00003	Инструкция 2	

**Корректная операция**



Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND	00001
00002	Инструкция 1	
00003	AND	00002
00004	Инструкция 2	

Если, как показано на диаграмме А, результат, существовавший в точке ветвления, не может измениться перед возвратом в точку ветвления (команды справа не влияют на результат), командная линия с ветвлением будет исполнена правильно и специальных мер программирования не требуется.

Если, как показано на диаграмме В, между точкой ветвления и “правосторонней” командой верхней командной линии существует условие, результат после выполнения верхней командной линии может измениться, тем самым сделав правильное выполнение нижней командной линии невозможным.

Для сохранения результатов при ветвлениях программ существует 2 метода:

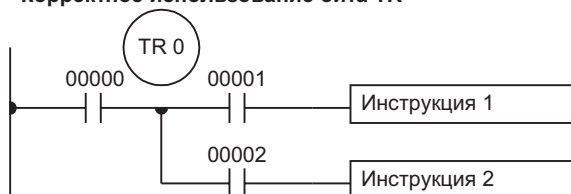
- использование битов TR
- использование секции INTERLOCK (IL/ IL).

**Биты TR**

Область TR имеет 8 битов, TR 0..TR 7, которые можно использовать для временного хранения результатов. Если TR поставлен в точке ветвления, текущее условие исполнения будет сохранено в указанном бите TR. При возвращении в точку ветвления бит TR возвращает условие исполнения, которое было сохранено при первом приходе в точку ветвления.

Предыдущую диаграмму В можно записать в следующем виде для правильного исполнения. В мнемокоде условие исполнения в точке ветвления запоминается в бите TR, как в операнде команды OUTPUT. После выполнения команды 1 условие исполнения возвращается из TR как операнд команды LOAD.

**Корректное использование бита TR**

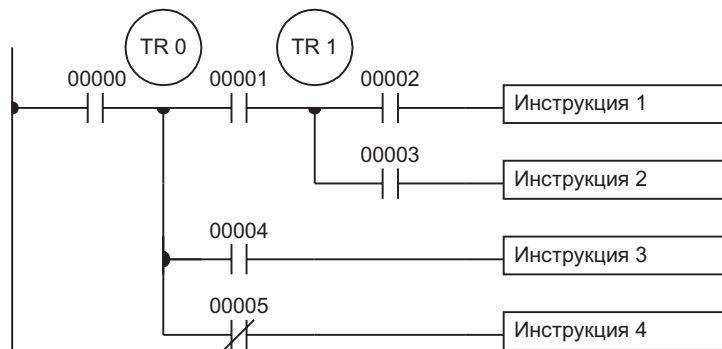


Адрес	Инструкция	Операнд
00000	LD	00000

Адрес	Инструкция	Операнд
00001	OUT	TR 0
00002	AND	00001
00003	Инструкция 1	
00004	LD	TR 0
00005	AND	00002
00006	Инструкция 2	

С точки зрения текущих команд вышеприведенная диаграмма работает следующим образом: загружается состояние IR 00000 (команда LOAD) для создания исходного условия исполнения. Условие исполнения в точке ветвления записывается в TR 0 командой OUTPUT. Далее производится операция И данного условия исполнения с IR 00001 и в соответствии с результатом выполняется команда 1. Сохраненное условие исполнения снова загружается (командой LOAD с TR 0 в качестве операнда), производится И с IR 00002 и в зависимости от результата выполняется команда 2.

Следующий пример показывает применение двух битов TR.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	OUT	TR 0
00002	AND	00001
00003	OUT	TR 1
00004	AND	00002
00005	Инструкция 1	
00006	LD	TR 1
00007	AND	00003
00008	Инструкция 2	
00009	LD	TR 0
00010	AND	00004
00011	Инструкция 3	
00012	LD	TR 0
00013	AND NOT	00005
00014	Инструкция 4	

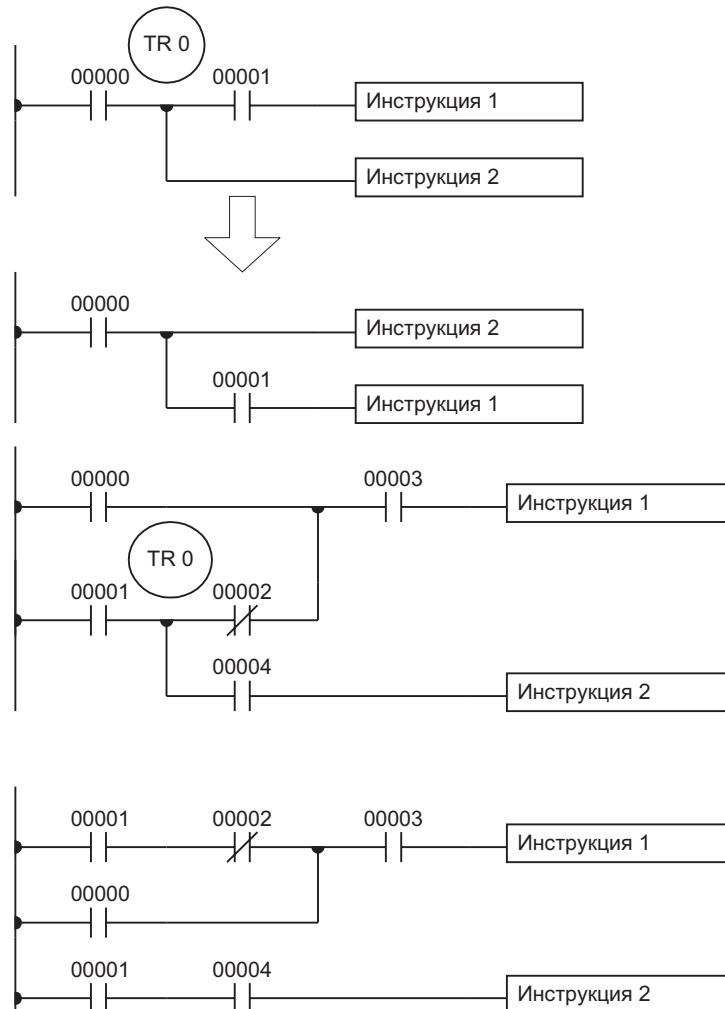
В данном примере TR 0 и TR 1 используются для сохранения условий исполнения в точках ветвления. После выполнения команды 1 условие исполнения, сохраненное в TR 1, загружается перед AND с IR 00003. Условие исполнения, сохраненное в TR 0, загружается 2 раза: первый раз для AND с IR 00004 и второй раз для AND с инверсии IR 00005.



Биты TR можно использовать столько раз, сколько необходимо, пока один и тот же бит TR не используется в одном модуле команд. Здесь, новый Модуль команд начинается каждый раз, когда происходит возврат на шину. Если в одном модуле команд необходимо иметь более 8 точек ветвления, которые требуют сохранения результата, нужно применять команды INTERLOCK (описаны далее).

При написании релейно-контактной схемы старайтесь не пользоваться битами TR без необходимости. Часто число команд, требуемых для программы, можно сократить и облегчить понимание программы путем преобразования релейно-контактной схемы, которые до преобразования потребовали битов TR. В обеих следующих парах диаграмм нижний вариант требует меньше команд и не требует битов TR. В первом примере это достигается реорганизацией части модуля команд: нижний путем отделения второй команды OUTPUT и использованием другой команды LOAD для создания условия исполнения.

**Замечание** Хотя упрощение программы доставляет хлопоты, иногда важен порядок выполнения. Например, команда MOVE может потребоваться перед командой двоичное сложение (BINARY ADD) для сохранения требуемых данных в требуемый операнд. Обязательно рассмотрите порядок исполнения перед упрощением программы.



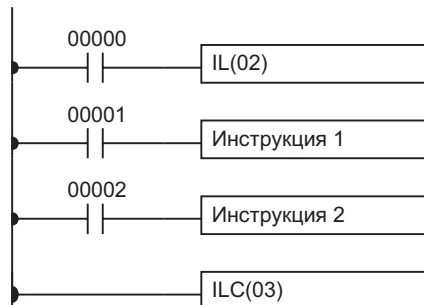
**Замечание** Биты TR должны вводиться только при программировании в мнемокоде. При вводе релейно-контактных схем их вводить не нужно, поскольку это делается автоматически. Однако вышеуказанные ограничения на число ветвлений, требующих TR, и на методы сокращения команд сохраняются.

**Секция INTERLOCK**

Проблемы сохранения условий исполнения в точках ветвления можно решить также командами INTERLOCK (сгруппировать) IL и INTERLOCK CLEAR (разгруппировать) ILC для полного устранения точек ветвления, но позволив указанным условиям исполнения управлять группами команд. Команды INTERLOCK и INTERLOCK CLEAR всегда используются совместно.

Когда команда INTERLOCK помещена перед участком релейно-контактной программы, условие исполнения для команды INTERLOCK управляет исполнением всех команд до команды INTERLOCK CLEAR. Если условие исполнения для команды INTERLOCK = 0, все выходные (“правосторонние”) команды до INTERLOCK CLEAR будут выполняться с условием 0 (т.е. сброс всей секции релейно-контактных схем). Влияние, которое они оказывают на конкретные команды, смотрите гл. 5.10 INTERLOCK и INTERLOCK CLEAR.

Диаграмму В можно откорректировать также с помощью INTERLOCK. Здесь условия исполнения до точки ветвления ставятся в командную строку для команды INTERLOCK. все строки от точки ветвления записываются как отдельные командные строки и добавляется еще одна команда INTERLOCK CLEAR. Обратите внимание, что ни INTERLOCK, ни INTERLOCK CLEAR не требуют операнда.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	IL	-
00002	LD	00001
00003	Инструкция 1	
00004	LD	00002
00005	Инструкция 2	
00006	ILC	-

Если в новой версии диаграммы В IR 00000 = 1, состояние IR 00001 и IR 00002 будут определять условие исполнения команд 1 и 2. Поскольку IR 00000 = 1, результат будет таким же, как и И с состоянием каждого из этих битов. Если IR 00000 = 0, команда INTERLOCK передаст значение 0 командам 1 и 2 и далее исполнение программы продолжится до строки с командой INTERLOCK CLEAR.

Как показано на следующей диаграмме, внутри модуля можно применить более одной команды INTERLOCK. Каждая действует до первой INTERLOCK CLEAR.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	IL	-
00002	LD	00001
00003	Инструкция 1	
00004	LD	00002
00005	IL	-
00006	LD	00003
00007	AND NOT	00004
00008	Инструкция 2	
00009	LD	00005
00010	Инструкция 3	
00011	LD	00006
00012	Инструкция 4	
00013	ILC	-

Если в данной диаграмме IR 00000 = 0 (т.е. условие исполнение для первой команды INTERLOCK = 0), команды 1..4 будут выполнены с условием исполнения 0 и далее произойдет переход к команде, следующей за INTERLOCK CLEAR. Если в данной диаграмме IR 00000 = 1, состояние IR 00001 будет загружено как условие исполнения команды 1, затем состояние IR 00001 будет загружено как условие исполнения второй команды INTERLOCK. Если IR 00002 = 0, команды 2..4 будут исполнены с условием 0. Если IR 00002 = 1, IR 00003, IR 00005, IR 00006 определяют первое условие исполнения на командных строках.

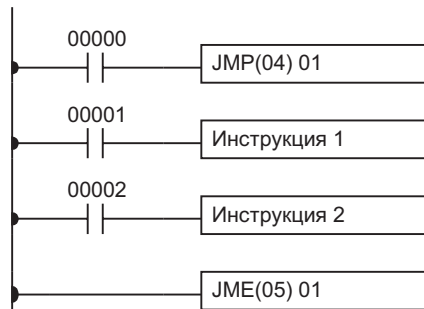
#### 4.7.8 Переходы

Заданную секцию программу можно пропустить в зависимости от заданных условий исполнения. Хотя это похоже на тот случай, когда условие исполнения для команды INTERLOCK = 0, при JUMP операнды всех условий сохраняют состояние. Переходами можно пользоваться для управления устройствами, которым требуется стабильный выход, т.е. пневматика и гидравлика, в то время как INTERLOCK можно использовать для управления устройств, которые не требуют установившегося выхода, напр. электронные устройства.

Переходы создаются командами JMP и JME (Конец перехода). Если условие исполнения для команды JUMP = 1, программа выполняется, как будто данной команды не существует. Если условие исполнения для команды JUMP = 0, программа переходит сразу к команде JUMP END, не меняя состояний выходов между JUMP и JUMP END.

Всем командам JUMP и JUMP END присвоены номера 00..99. Есть 2 типа переходов. Номер перехода определяет тип перехода.

Переход можно определить номерами 01..99 только один раз, т.е. каждый номер может использоваться один раз с командой JUMP и один раз JUMP END. Когда выполнена команда JUMP, сразу происходит переход к JUMP END с тем же номером, как будто команд JUMP с другими номерами не существуют. Диаграмма В из примера с TR и INTERLOCK может быть переписана с использованием команд JUMP, как показано ниже. Хотя в качестве номера перехода используется номер 01, можно использовать любой номер 00..99, если они не используются в других местах программы. JUMP и JUMP END не требуют операндов, а JUMP END не имеет и условий исполнения в командной строке.



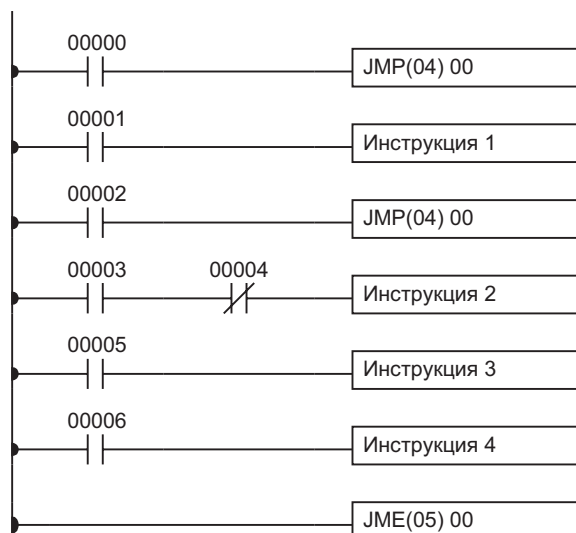
Адрес	Инструкция	Операнд
00000	LD	00000
00001	JMP	01
00002	LD	00001
00003	Инструкция 1	
00004	LD	00002
00005	Инструкция 2	
00006	JME	01

Данная версия диаграммы В будет иметь самое короткое время исполнения из всех, если IR 00000 = 0.

Другой тип перехода создает JUMP с номером 00. С помощью JUMP с номером 00 можно создать сколь угодно много команд, и JUMP с номером 00 можно использовать без JUMP END между ними. Возможно, даже для всех команд JUMP 00 перенести исполнение программы к одному JUMP END 00, т.е. для всех команд JUMP 00 требуется только одна команда JUMP END 00. Когда в качестве номера перехода задан 00, Исполнение программы переходит к ближайшей команде JUMP END 00. Хотя, как и при всех переходах, между JUMP 00 и JUMP END 00 состояние не изменяется и команды не выполняются, программа должна искать ближайшую команду JUMP END 00, чем немного продлевается время исполнения.

Исполнение программы с несколькими командами JUMP 00 и одной JUMP END 00 аналогично исполнению программы с командой INTERLOCK. Следующая диаграмма аналогична той, которая исполнялась с командой INTERLOCK. но переписана с командами JUMP.

Выполнение программы будет отличаться, (например, в предыдущей сбрасывались некоторые секции программы, а переходы не влияют на состояние битов между JUMP и JUMP END).



Адрес	Инструкция	Операнд
00000	LD	00000
00001	JMP	00
00002	LD	00001
00003	Инструкция 1	
00004	LD	00002
00005	JMP	00
00006	LD	00003
00007	AND NOT	00004
00008	Инструкция 2	
00009	LD	00005
00010	Инструкция 3	
00011	LD	00006
00012	Инструкция 4	
00013	JME	00

## 4.8 Управление состоянием битов

Есть 5 базовых команд, используемых для управления состоянием отдельного бита. Это OUTPUT, OUTPUT NOT, DIFFERENTIATE UP, DIFFERENTIATE DOWN и KEEP. Все эти команды ставятся самыми последними в командной линии и требуют битового адреса в качестве операнда. Хотя подробности приведены в гл. 5, данные команды (за исключением OUTPUT и OUTPUT NOT, которые уже описаны) описаны здесь из-за их важности в большинстве программ. Хотя данные команды используются для переключения в 0 или 1 выходные биты области IR (т.е. выдать или убрать выходной сигнал с внешнего устройства), их также можно использовать для управления состоянием других битов в области IR или в других областях.

### 4.8.1 Включить на 1 цикл (DIFFERENTIATE UP и DIFFERENTIATE DOWN)

Команды DIFFERENTIATE UP (включить на цикл при условии 0/1) и DIFFERENTIATE DOWN (включить на цикл при условии 1/0) используются для включения битового операнда в 1 на 1 цикл. Команда DIFFERENTIATE UP включает операнд в состояние 1

после того, как условие исполнения изменился с 0 на 1; Команда DIFFERENTIATE DOWN.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	DIFU	00201



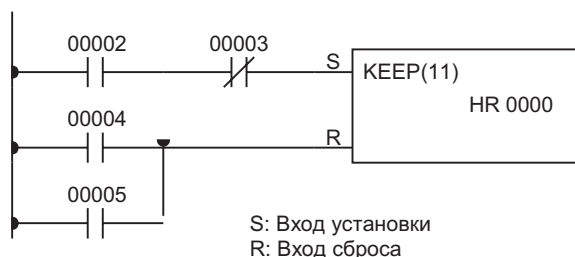
Адрес	Инструкция	Операнд
00000	LD	00001
00001	DIFU	00201

Здесь, IR 00200 будет включен в 1 на 1 цикл после включения IR 00000 в состояние 1. На следующем цикле после выполнения DIFU IR 00200 будет выключено в 0 независимо от состояния IR 00000. Командой DIFFERENTIATE DOWN IR 00201 будет включен в 1 на 1 цикл после включения IR 00001 в состояние 0 (до того времени IR 00201 будет = 0), и будет выключен в 0 после выполнения DIFD.

#### 4.8.2 Сохранить (KEEP)

Команда KEEP служит для сохранения состояния битового операнда и требует двух входов условий исполнения. К команде KEEP поведятся две командные линии. Когда условие исполнение на первой линии = 1, битовый операнд становится = 1.. Когда условие исполнения на второй линии = 1, битовый операнд становится = 0. Битовый операнд команды KEEP будет сохранять состояние 0 или 1, даже если он находится на секции диаграммы с командой INTERLOCK.

В следующем примере HR 0000 будет включен в 1, когда IR 00002 = 1 и IR 00003 = 0. HR 0000 останется в состоянии 1 до тех пор, когда или IR 00004 или IR 00005 включатся в 1. Как у всех команд, требующих более одной командной строки, командные строки кодируются перед командой, которой они управляют.



Адрес	Инструкция	Операнд
00000	LD	00002
00001	AND NOT	00003
00002	LD	00004
00003	OR	00005
00004	KEEP	HR 0000

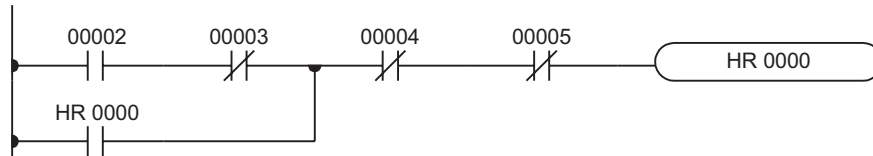
#### 4.8.3 Биты самоподдержки

Хотя команду KEEP можно использовать для создания битов самоподдержки. иногда необходимо создать такие биты другим образом, чтобы их можно было выключить при работе с командой INTERLOCK.

Для создания битов самоподдержки служит битовый операнд команды OUTPUT, который сам является условием исполнения для той же OUTPUT (подключен параллельно),

чтобы битовый операнд команды OUTPUT оставался в состоянии 1 или 0 до тех пор, пока не произойдет переключение других битов. Как минимум еще одно условие должно использоваться перед командой OUTPUT для сброса. Без этого условия сброса управлять Битовым операндом невозможно.

Вышеприведенную диаграмму с командой KEEP можно переписать, как показано на диаграмме ниже. Единственная разница в диаграммах будет только при работе в группе с командой INTERLOCK, когда условие команды INTERLOCK = 1. Здесь, как и на диаграмме с командой KEEP, используются 2 бита сброса, т.е. HR 0000 можно переключить в 0 включением в 1 либо IR 00004, либо IR 00005.



Адрес	Инструкция	Операнд
00000	LD	00002
00001	AND NOT	00003
00002	OR	HR 0000
00003	AND NOT	00004
00004	OR NOT	00005
00005	OUT	HR 0000

#### 4.9 Рабочие биты (внутренние реле)

Часто трудно запрограммировать условия так, чтобы сразу получить условия исполнения. Эти трудности легко преодолеваются тем, что используются определенные биты для переключения других команд. Такое программирование достигается с помощью рабочих битов. Иногда для этих целей требуются целые слова. Такие слова называются рабочими.

Рабочие биты не передаются с или на ПК. Они выбраны программистом для облегчения программирования. Биты входов/выходов и другие биты специального назначения нельзя использовать в качестве рабочих битов. Все биты в области IR, не выделенные в качестве битов входов/выходов, и некоторые неиспользуемые биты зоны AR можно использовать в качестве рабочих битов. Это облегчает разработку и написание программы, а также отладку.

##### Применение рабочих битов

Примеры в данном пункте показывают два основных способа применения рабочих бит. Они могут служить примером для почти неограниченного числа способов применения рабочих бит. Если возникают проблемы при программировании, нужно рассмотреть возможность применения рабочих бит для упрощения программирования.

Рабочие биты часто используются с командами OUTPUT, OUTPUT NOT, DIFFERENTIATE UP, DIFFERENTIATE DOWN и KEEP. Сначала рабочие биты используются в качестве операндов для этих команд, чтобы позднее использовать их в качестве условий для выполнения команд. Рабочие биты можно также использовать и с другими командами, напр. SHIFT REGISTER (регистр сдвига) (SFT). Пример использования рабочих слов с командой. SHIFT REGISTER (SFT) приведен в п. 5.15.1.

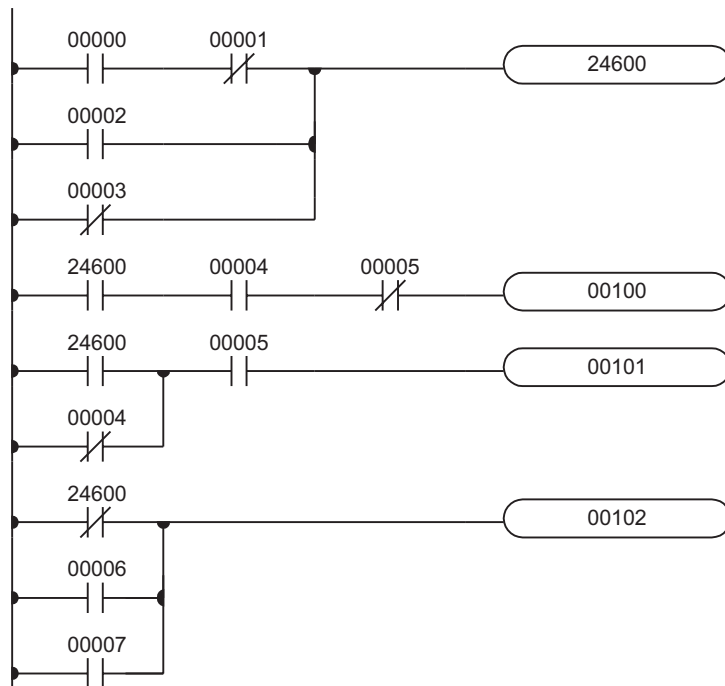
Много примеров применения рабочих битов приведено в гл. 5, хотя они и не выделяются особо, как рабочие биты. Понимание применения рабочих бит важно для эффективного программирования.

##### Сокращение сложных условий

Рабочие биты можно использовать для упрощения программы, когда некоторая комбинация условий часто используется с другими условиями. В следующем примере IR 00000, IR 00001, IR 00002 и IR 00003 объединяются в логический Модуль, который

#### 4.9 Рабочие биты (внутренние реле)

сохраняет результирующее условие исполнение в IR 216000. Далее IR 21600 объединяется в логические модули с различными другими условиями для задания условий срабатывания IR 10000, IR 10001 и IR 10002, т.е. включение выходов, приписанных к этим битам, в 1 или 0.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND NOT	00001
00002	OR	00002
00003	OR NOT	00003
00004	OUT	24600
00005	LD	24600
00006	AND	00004
00007	AND NOT	00005
00008	OUT	00100
00009	LD	24600
00010	OR NOT	00004
00011	AND	00005
00012	OUT	00101
00013	LD NOT	24600
00014	OR	00006
00015	OR	00007
00016	OUT	00102

#### Условия при работе с фронтами

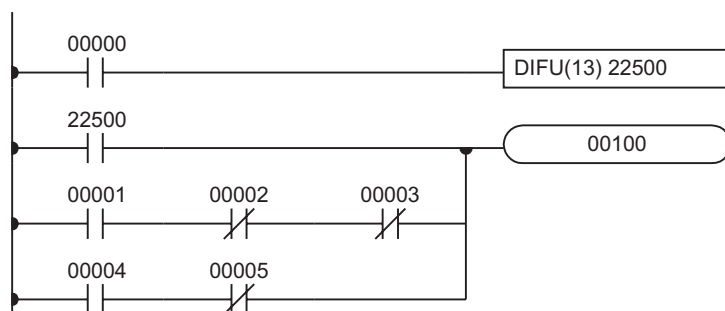
Рабочие биты можно также использовать для работы с фронтами для некоторых, но не для всех условий, требуемых для выполнения команд. В данном примере IR 10000 должен оставаться 1, пока IR 00001 = 1, а IR 00002 и IR 00002 = 0, или пока IR 00004 = 1 и IR 00005 = 0. Он должен будет = 1 только на 1 цикл каждый раз, когда IR 00000 включается в 1 (если только одно из предыдущих условий не находится постоянно в состоянии 1).

Это легко запрограммировать применением IR 22500 в качестве рабочего бита как операнда для команды DIFFERENTIATE UP (DIFU). Когда IR 10000 включается в 1, IR



#### 4.10 Предосторожности при программировании

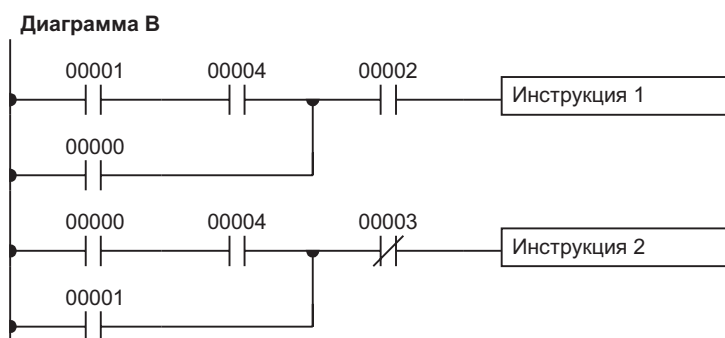
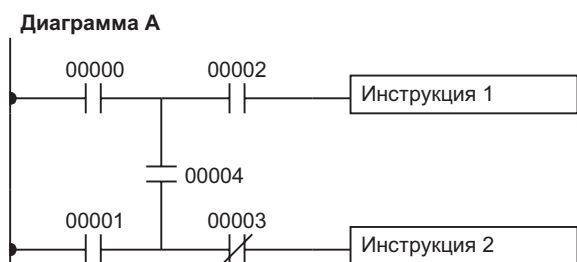
22500 включится в 1 на время 1 цикла и в следующем цикле сбросится в 0 командой DIFU. Если другие условия, управляющие IR 10000, не 1, рабочий бит IR 22500 включит на время одного цикла IR 10000 в 1.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	DIFU	22500
00002	LD	22500
00003	LD	00001
00004	AND NOT	00002
00005	AND NOT	00003
00006	OR LD	-
00007	LD	00004
00008	AND NOT	00005
00009	OR LD	-
00010	OUT	00100

#### 4.10 Предосторожности при программировании

Число условий, которые можно применять последовательно или параллельно, ограничен только емкостью памяти ПК. Так что используйте столько условий, сколько требуется для написания понятную диаграмму. Хотя с помощью командных линий можно написать очень сложные диаграммы, не должно быть условий на вертикальных линиях между двумя командными линиями. Диаграмма А недопустима, ее нужно перерисовать так, как представлено на диаграмме В. Мнемокоды даны только для диаграммы В; кодирование диаграммы А невозможно.

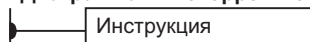


Адрес	Инструкция	Операнд
00000	LD	00001
00001	AND	00004
00002	OR	00000
00003	AND	00002
00004	Инструкция 1	
00005	LD	00000
00006	AND	00004
00007	OR	00001
00008	AND NOT	00003
00009	Инструкция 2	

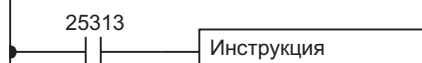
Бит может присваиваться условиям неограниченное число раз, так что используйте их столько раз, сколько требуется для упрощения программы. Часто сложные программы получаются из-за попыток сократить число применений бита.

За исключением команд, для которых условия не разрешены (напр. INTERLOCK CLEAR или JUMP END, см. ниже) каждая командная линия должна иметь как минимум одно условие для задания условия исполнения выходной (“правосторонней”) команды. Диаграмма А должна быть переписана в виде диаграммы В. Если команда должна выполняться непрерывно (например, выход должен быть постоянно 1 при выполнении программы), используйте флаг Всегда 1 (SR 25313).

**Диаграмма А: некорректно**



**Диаграмма В**

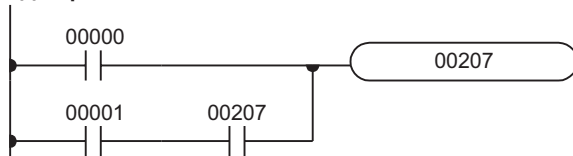


Адрес	Инструкция	Операнд
00000	LD	25313
00001	Инструкция	

Есть несколько исключений из этого правила, включая INTERLOCK CLEAR, JUMP END и команды секции STEP. Каждая из этих команд используется как вторая команды из пары, и управляется условием исполнения первой команды пары. На командных линиях, ведущих к этим командам, не должно быть условий. Подробности см. гл. 5, Команды.

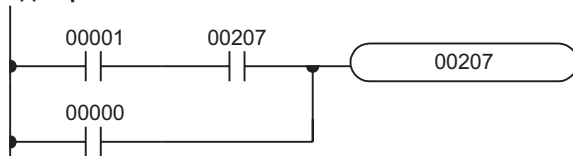
При написании релейно-контактных схем всегда нужно помнить о количестве команд, которые потребуются для ее ввода. На диаграмме А потребуется команда OR LOAD для объединения верхней и нижней командной линий. Этого можно избежать, переписав ее в виде диаграммы В, в которой AND LOAD или OR LOAD не потребуется. Подробности см. 5-8-2 и дополнительные примеры в гл. 7.

Диаграмма А



Адрес	Инструкция	Операнд
00000	LD	00000
00001	LD	00001
00002	AND	00207
00003	OR LD	-
00004	OUT	00207

Диаграмма В



Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND	00207
00002	OR	00001
00003	OUT	00207

## 4.11 Исполнение программы

Когда начато исполнение программы, ЦПУ исполняет программу, начиная с первой строки и до последней, проверяя все условия и выполняя все команды по мере движения сверху вниз. Важно располагать команды в нужном порядке, чтобы данные заносились в слово перед тем, как использовать его в качестве операнда. Помните, что сначала выполняется командная линия к “правосторонней” (выходной) команде, а потом командные линии ветвления, ведущие к другим “правосторонним” (выходным) командам. Исполнение программы - только часть задач, которые ЦПУ выполняет за время цикла. Подробности см. гл. 6.

## 4.12 Программы работы с модулями специальных входов/выходов

В данной главе описаны методы программирования и меры предосторожности при операциях с модулями специальных входов/выходов.

### 4.12.1 Перезапуск модулей специальных входов/выходов

При перезапуске модулей специальных входов/выходов запрещена команда IOFR до завершения инициализации модулей специальных входов/выходов.

## 4.12 Программы работы с модулями специальных входов/выходов



Пока флаг перезапуска SR 27400 = 1, осуществляется нормальное обновление END и инициализируется Модуль специальных входов/выходов. Это происходит независимо от значений DM 6620 и DM 6621, которые относятся к обновлению специальных входов/выходов. Команда IOFR в программе не будет исполняться, пока не завершится инициализация модуля.

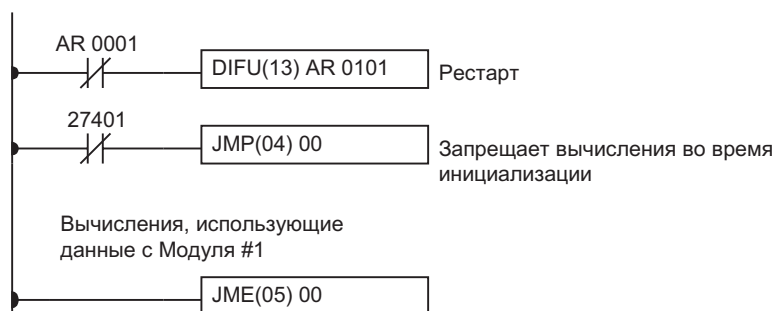
Параметры модуля специальных входов/выходов, которые должны обновляться, могут быть потеряны при инициализации. При написании программы для перезапуска модулей специальных входов/выходов запретите работу программ, которые зависят от данных, получаемых от инициализируемых модулей специальных входов/выходов, такие, как данные, используемые в вычислениях, пока флаг перезапуска (SR 27400..SR 27415) = 1. Нормальные операции можно проводить для модулей, которые не инициализируются.

Флаг рестарта не будет включаться в 1 для модулей специальных входов/выходов, установленных на ведомых панелях.

Стандартное время перезапуска специальных входов/выходов = 20 x время цикла.

### 4.12.2 Программа обработки ошибок модулей специальных входов/выходов

Для рестарта модуля специальных входов/выходов, в котором произошла ошибка, пользуйтесь следующей программой. Данная программа перезапускает Модуль 1.



AR 0001: Флаг ошибки Модуля #1  
SR 27401: Флаг рестарта Модуля #1

### 4.12.3 Изменение параметров модулей специальных входов/выходов

В C200HX/HG/HE релейно-контактные команды можно использовать для записи данных в области модулей специальных входов/выходов (DM 1000..DM 2599) и изменения параметров модулей специальных входов/выходов. Изменение параметров полезно, когда требуются различные параметры для разных производственных процессов.

В данном примере есть 2 производственных процесса, которые требуют разных параметров модулей специальных входов/выходов. Параметры для первого процесса хранятся в DM 7000..DM 7999. Параметры для второго процесса хранятся в DM 8000..DM 8999.

#### Операции программатора

Шаги 1..5 в следующем алгоритме не являются необходимыми, когда XFER используется для перезаписи DM 1000..DM 1999 прямо из программы содержимым фиксированных DM (DM 6144..DM 6599). В этом случае просто перезапустите Модуль из программы после перезаписи DM 1000..DM 1999.

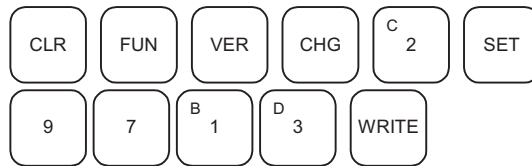
**1, 2, 3,...** 1. Сделайте полную очистку памяти.



Операцию выделения области UM нельзя проводить, не очистив память.

## 4.12 Программы работы с модулями специальных входов/выходов

2. Проведите операцию выделения области UM для выделения 2 К слов для области расширенных DM (DM 7000..DM 8999).

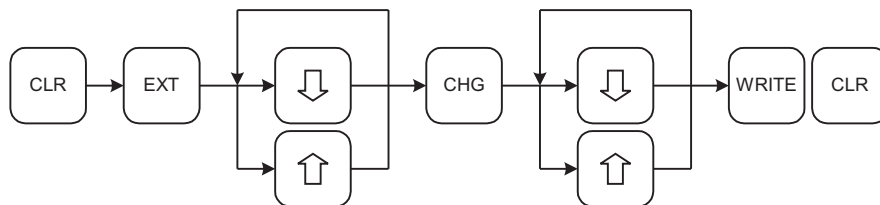


3. Проведите операцию модификации данных

16-ричные/двоично-десятичные для установки модуля специальных входов/выходов в режим "С200Н-совместимый режим 1 ROM" путем послылки в DM 6602 числа #0100. Этот режим передает содержимое DM 7000..DM 7999 в DM 1000..DM 1999 при пуске ПК. Эти новые параметры вступят в действие только перезапуска ПК путем выключения и повторного включения питания.

4. Установите секцию 4 переключателя DIP на ЦПУ в ON. Это позволяет пользователю присваивать дополнительным командам функциональные коды.

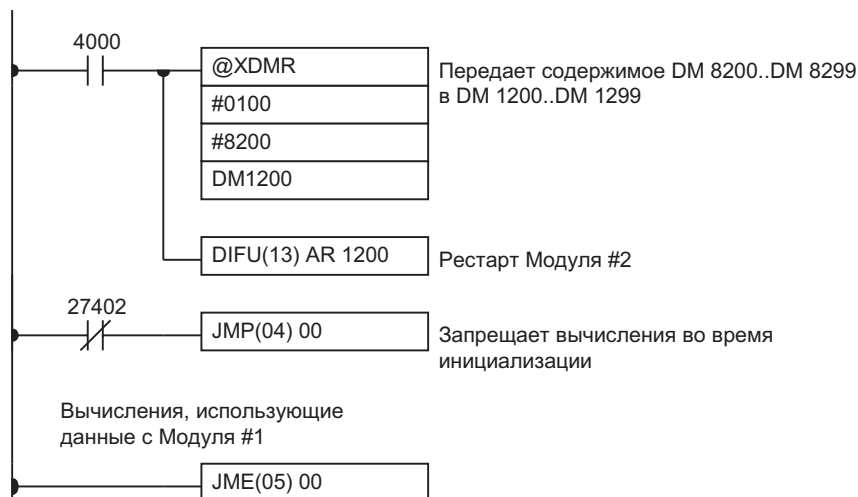
5. Проведите операцию присвоения дополнительным командам функциональных кодов для присвоения функционального кода команде XDMR.



6. Введите программу.

### Пример программы (Модуль 2 специальных входов/выходов)

Следующая программа изменяет установочные параметры области модуля специальных входов/выходов для модуля 2, перезапускает Модуль и запрещает вычисления, использующие данные с модуля 2, во время инициализации модуля.



### 4.12.4 Интервал обновления входов/выходов модуля специальных входов/выходов

Когда интервал между обновлением входов/выходов слишком короток, обработка модуля специальных входов/выходов может быть задержано, что вызовет ошибки модуля специальных входов/выходов или иным образом мешая нормальной работе

## 4.12 Программы работы с модулями специальных входов/выходов

модуля. В этом случае используйте следующие методы для возобновления нормальной работы.

### Короткие интервалы между обновлениями END

Есть два способа удлинения интервалов между обновлениями входов/выходов. Можно применять любой метод.

- 1, 2, 3,...** 1. Запретить циклические обновления модуля специальных входов/выходов установочным параметром DM 6621 и использовать IOFR для обновления входов/выходов специальных модулей только когда необходимо.

Для запрещения циклических обновлений модуля специальных входов/выходов, установленного на панели ЦПУ или панелях расширения входов/выходов задайте #0100 в DM 6621.

2. Увеличить время цикла ПК заданием нового времени цикла в параметре DM 6619 или выполнив команду SCAN. (SCAN - это команда расширения с кодом по умолчанию 18).

### Короткий интервал между IOFR и обновлениями END

Измените программу для использования либо обновления IOFR, либо обновления END. Также возможно увеличить время цикла ПК заданием времени минимального цикла в параметре DM 6619 или выполнив команду SCAN в программе.

### Короткий интервал между командами IOFR

Измените программу для увеличения времени между командами IOFR или просто использование одной команды IOFR.

#### 4.12.5 Сокращение времени цикла

Когда на C200HX/HG/HE установлен Модуль специальных входов/выходов, обновление END осуществляется автоматически каждый цикл без установки параметров. Когда используются несколько модулей специальных входов/выходов, время цикла может стать слишком долгим из-за времени, требующемуся для автоматического обновления модулей специальных входов/выходов.

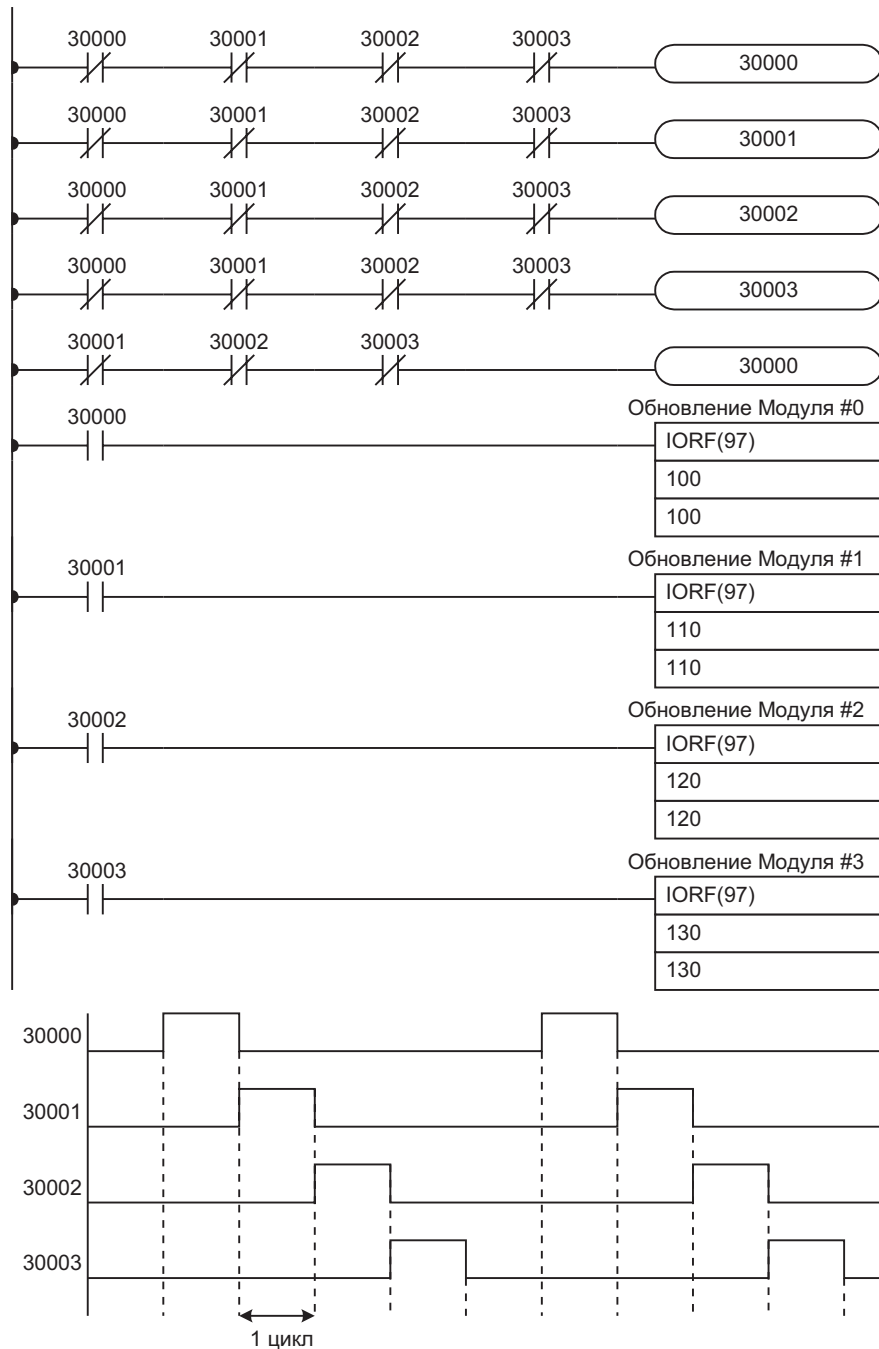
Для сокращения времени, отведенному на обновление модулей специальных входов/выходов, запретите циклическое обновление модулей специальных входов/выходов в параметре DM 6621 и вместо этого используйте IOFR для обновления модулей специальных входов/выходов. Обновление остальных модулей специальных входов/выходов, установленных на панели ЦПУ или панелях расширения входов/выходов можно запретить установочным параметром, пошлав #0100 в DM 6621.

Следующий пример программы сокращает время обновления модулей специальных входов/выходов для ПК с 4 модулями специальных входов/выходов путем обновления одного модуля за цикл. Модули обновляются в порядке: Модуль 0, Модуль 1, Модуль 2, Модуль 3, Модуль 0 и т. д.

Модуль #0	Модуль #1	Модуль #2	Модуль #3		ЦПУ
-----------	-----------	-----------	-----------	--	-----

Следующий пример программы относится только к модулям специальных входов/выходов, установленных на панели ЦПУ или панели расширения входов/выходов, поскольку обновление END всегда осуществляется для модулей специальных входов/выходов, установленных на ведомой панели, независимо от значений параметров.

## 4.13 Программирование модуля аналогового таймера



**Замечание** IR 30000 используется для команды OUT дважды в программе. Хотя данный тип дублирования приемлем в этом примере, он обычно запрещен, за исключением случаев, когда есть особые причины для применения и затем можно обеспечить корректную работу.

## 4.13 Программирование модуля аналогового таймера

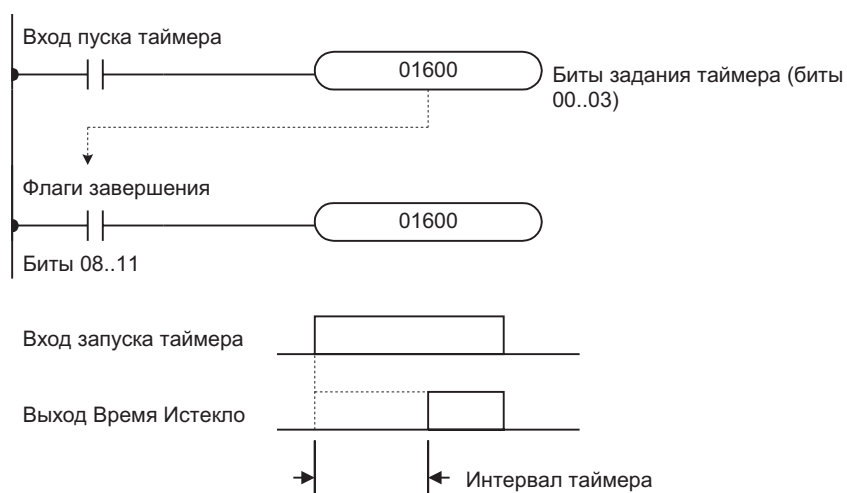
Задание модуля аналогового таймера можно легко изменять без программатора. На модуле есть разъем для подключения внешнего переменного резистора, так что на панель управления можно установить внешний переменный резистор и подключить его к блоку аналогового таймера для ручного задания и настройки задания таймера.

### 4.13.1 Работа

Когда вход пуска таймера = 1, биты установки таймера, выделенные блоку аналогового таймера (биты 00..03 n), устанавливаются в 1, аналоговый таймер начинает работу, и индикатор задания таймера (SET) на модуле горит.

Когда задание таймера (SV) (установленное извне или изнутри) истекает, флаги завершения модуля (биты 08..11 n) и выход ВРЕМЯ ИСТЕКЛО будут установлены в 1. Кроме того, будет гореть индикатор на модуле ВРЕМЯ ИСТЕКЛО.

Подробности о переключении между внутренним и внешним заданием таймера, подключении переменного резистора и установках переключателя см. Инструкцию по работе модулей аналогового таймера.



### 4.13.2 Распределение битов и установка переключателя DIP

В таблице показано использование слова (n), выделенного блоку аналогового таймера. Адрес модуля зависит от платоместа, в котором установлен Модуль.

Бит	Класс (вход или выход)	Функция	Комментарии
00	Выход	Бит установки таймера 0	1, когда таймер задан
01		Бит установки таймера 1	
02		Бит установки таймера 2	
03		Бит установки таймера 3	
04		Бит останова таймера 0	0: разрешение работы таймера
05		Бит останова таймера 1	
06		Бит останова таймера 2	1: Останов работы таймера
07		Бит останова таймера 3	
08	Вход	Флаг завершения таймера 0	1, когда таймер отработан
09		Флаг завершения таймера 1	
10		Флаг завершения таймера 2	
11		Флаг завершения таймера 3	
12..15	-	Не используется	-

#### Задание диапазона таймера

Задаёт диапазон таймера верхним переключателем DIP на передней панели модуля. Диапазон каждого таймера можно задать независимо.



### 4.13 Программирование модуля аналогового таймера

Диапазон	Таймер 0		Таймер 1		Таймер 2		Таймер 3	
	Секция							
	8	7	6	5	4	3	2	1
0.1..1 с	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
1..10 с	ON	OFF	ON	OFF	ON	OFF	ON	OFF
10..60 с	OFF	ON	OFF	ON	OFF	ON	OFF	ON
11..10 мин	ON	ON	ON	ON	ON	ON	ON	ON

#### Выбор задания внутреннее/внешнее

Выбирайте внутреннее или внешнее задание нижним переключателем DIP на передней панели модуля.

Источник	Таймер 0 (Секция 4)	Таймер 1 (Секция 3)	Таймер 2 (Секция 2)	Таймер 3 (Секция 1)
Внутреннее	ON	ON	ON	ON
Внешнее	OFF	OFF	OFF	OFF

#### 4.13.3 Пример программы

##### Конфигурация модуля

В следующей таблице показано распределение слов для модулей в примере.

Назначение слова	Слово
Слово IR, выделенное блоку аналогового таймера	IR 002
Слово IR, выделенное блоку входов	IR 000
Слово IR, выделенное блоку выходов	IR 005

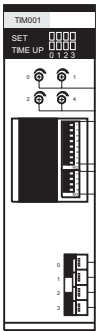
Значения задания модуля аналогового таймера и подключение внешнего резистора показаны ниже.

Таймер	Задание	Диапазон	Значение переменного резистора	Управление переменным резистором
0	0.6 с	0.1..1 с	60% по ЧС	Внутренне
1	3 с	1..10 с	30% по ЧС	Внутренне
2	20 с	10..60 с	20% по ЧС	Внешнее
3	8 минут	1..10 мин	80% по ЧС	Внешнее

##### Параметры и подключение модуля

На следующей схеме показаны положения переключателя и подключения, требуемые для создания ранее приведенной конфигурации.

## 4.13 Программирование модуля аналогового таймера



Задание для этих двух переменных резисторов действует, так как таймеры 0 и 1 заданы для внутреннего задания. Для настройки резисторов применяйте отвертку, поставляемую с Модулем.

Задание для этих двух переменных резисторов не действует, так как таймеры 2 и 3 заданы для внешнего задания.

Диапазон таймера

Таймер 0	Таймер 1	Таймер 2	Таймер 3				
0.1..1 с	1..10 с	10..60 с	1..10 мин				
Секция							
8	7	6	5	4	3	2	1
OFF	OFF	ON	OFF	OFF	ON	ON	ON

Тип источника задания (внешний/внутренний)

Таймер 0	Таймер 1	Таймер 2	Таймер 3
Секция 4	Секция 3	Секция 2	Секция 1
ON	ON	OFF	OFF
Внутренний	Внутренний	Внешний	Внешний

Не подключайте ничего к этим разъемам. Таймеры 0 и 1 настроены на внешнее задание.

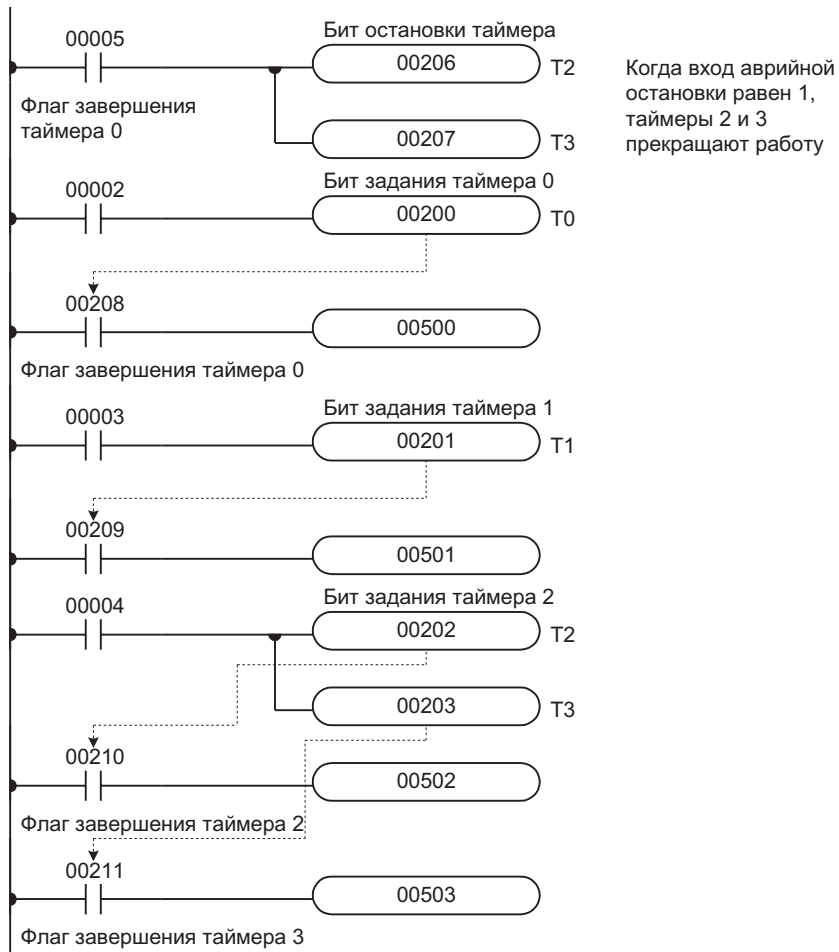
Параметры внешнего задания.  
К данным разъемам подключите переменные резисторы задания таймеров 2 и 3. Подробности см. Инструкцию по работе Модуля Аналогового Таймера.

### Релейно-контактная программа

Далее показан пример релейно-контактной схемы

- 1, 2, 3,...**
- Выход IR 00500 станет = 1 спустя около 0.6 с (T0) после включения в 1 входа IR 00002.
  - Выход IR 00501 станет = 1 спустя около 3 с (T1) после включения в 1 входа IR 00003.
  - Выход IR 00502 станет = 1 спустя около 20 с (T2) после включения в 1 входа IR 00004 и IR 00503 станет = 1 спустя около 8 минут (T3) после включения в 1 входа IR 00004
  - Таймеры 2 и 3 останавливаются входом IR 00005.

### 4.13 Программирование модуля аналогового таймера





## 5. Набор команд

*S200HX/HG/HE* располагает большим числом команд программирования, позволяющих просто запрограммировать сложные процессы управления. В данной главе объясняется каждая команда и приводятся ее символ и изображения на релейно-контактной схеме, области памяти и флаги, используемые с каждой командой.

*Команды S200HX/HG/HE* делятся на следующие группы: релейно-контактные команды, команды управления битами, команды таймеров и счетчиков, команды сдвига данных, команды сравнения, команды пересылки данных, команды преобразования данных, команды двоично-десятичных вычислений, команды двоичных вычислений, логические команды, подпрограммы, команды связи, дополнительные команды ввода/вывода и команды специальных модулей входов/выходов.

Некоторые команды, такие как команды таймера и счетчика, используются для управления исполнением других команд, например, Флаг завершения ТИМ, можно использовать для управления включением в 1 бита с помощью командой выхода, их можно также использовать для управления другими командами. Следовательно, для конкретных задач команды выхода, использованные в примерах данной инструкции, можно заменить другими командами, а не управлять непосредственно битами.

## 5.1 Общие сведения

В данной инструкции все обращения к командам будет производиться по их мнемонике. Например, команда OUTPUT будет названа OUT, команда AND LOAD - AND LD. Если Вы не уверены, какую команду означает мнемоника, см. Приложение В, команды программирования.

Если команде присвоен функциональный код, он будет дан в скобках после мнемоники. Данные функциональные коды, две двоично-десятичные цифры, используются для ввода команд в ЦПУ и кратко описаны далее, а более подробно в гл. 4.7. Таблица команд в порядке их функциональных кодов также приведена в Приложении В.

Знак @ перед мнемоникой указывает на версию данной команды "Команда ФРОНТА 0/1". Команды ФРОНТА 0/1 описаны в п. 5-4.

## 5.2 Формат команды

Большинство команд имеет один или несколько связанных с ней операндов. Операнд указывает или предоставляет данные, которые должна обрабатывать команда. В их качестве иногда выступают входы, с которых могут поступать текущие численные значения. (т.е. константы), но обычно это адреса слов или бит, которые содержат требуемые данные. Бит, адрес которого указан в качестве операнда, называется битовым операндом. Слово, адрес которого указан в качестве операнда, называется словным операндом. В некоторых командах слово, указанное в качестве операнда, указывает на первое из нескольких слов, содержащих требуемые данные.

Каждая команда требует один или несколько слов памяти программ. Первое слово - слово команды, которое задает команду и содержит определители (описанные ниже) или битовые операнды, требуемые командой. Другие операнды, требуемые командой, содержатся в следующих словах, один операнд на слово. Некоторые команды требуют до 4 слов.

Определитель - это операнд, связанный с командой и находящийся с одним словом с операндом. Такие операнды определяют команду, а не указывают на данные. Примеры определителей - номера таймеров/счетчиков, которые используются для создания таймеров и счетчиков, а также номера переходов (которые определяют пару команд JUMP и связанную с ней JUMP END). Битовые операнды также содержатся в одном слове с командой, хотя и не являются определителями.

## 5.3 Области данных, значения определителей и флаги

В данной главе каждое описание команды включает релейно-контактный символ, области данных, которые могут использоваться операндами, и значения, которые могут принимать определители. Подробности об областях данных уточняются также именами операндов и типом данных, требуемых каждым операндом (т.е. слово или бит, а для слова - 16-ричное или двоично-десятичное).

Не все адреса в указанных областях доступны для операндов, например, если операнд требует два слова, последнее слово в области данных не может назначаться первым словом операнда, поскольку все слова одного операнда должны находиться в одной области. Кроме того, не все слова в областях SR и DM можно записать в качестве операндов. (Подробности см. гл. 3, Области памяти). Дополнительная информация о специфических ограничениях дана в подразделе Ограничения. Об адресации и адресах флагов и битов управления см. гл. 3 Области памяти.

**Внимание!** Области IR и SR рассматриваются как отдельные области данных. Если операнд имеет доступ к одной области, это еще не значит, что этот же операнд будет иметь доступ к другой области. Граница между IR и SR для одного операнда может, однако, пересекаться., т.е. последний бит в области IR можно определить для операнда, требующего более одного слова, если область SR разрешена для данного операнда.

### 5.3 Области данных, значения определителей и флаги

Подраздел Флаги перечисляет флаги, на которые влияет команда. Эти флаги включают следующие флаги области SR

Сокращение	Название	Бит
ER	Instruction Execution Error Flag Флаг ошибки исполнения команды	25503
CY	Carry Flag Флаг переноса	25504
GR	Greater Than Flag Флаг больше чем	25505
EQ	Equals Flag Флаг равно	25506
LE	Less Than Flag Флаг меньше чем	25507
N	Negative Flag Флаг отрицательно	25402
OF	Overflow Flag Флаг переполнения +	25404
UF	Underflow Flag Флаг переполнения -	25405

ER наиболее часто используются для просмотра выполнения команды. Когда ER = 1, это указывает, что при выполнении текущей команды произошла ошибка. Раздел ФЛАГИ в описании каждой команды перечисляет возможные причины переключения ER в 1. ER включится в 1, когда операнды вводятся некорректно. Таблица команд и флагов, на которые они воздействуют, приведена в Приложении С.

#### Косвенная адресация

Когда в качестве операнда задана область DM, можно применять косвенную адресацию. Косвенная адресация задается звездочкой перед DM: \*DM.

Когда задана косвенная адресация, указанное слово DM содержит адрес слова DM, в котором содержатся данные, которые будут использоваться как операнд для команды. Если, например, для команды MOV в качестве первого операнда задан \*DM 0001, а в качестве второго LR 00, и содержимое DM 0001 было 1111, содержимое DM 1111 было 5555, значение 5555 будет передано в LR 00.



При использовании косвенной адресации адрес требуемого слова должно быть в двоично-десятичном виде и должен задавать слово в области DM. В указанном примере содержимое слова \*DM 00000 должно быть двоично-десятичным числом в диапазоне 0000..6655.

Команда IEMS используется для изменения адреса приемника \*DM из области DM на один из банков области EM. Подробности см. 5.25.15.

#### Обозначение констант

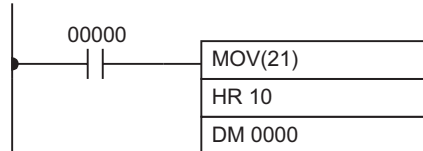
Хотя в большинстве случаев в качестве операндов задаются адреса областей, многие операнды и определители вводятся как константы. Диапазон значений определителя или операнда зависит от конкретной команды, которая их использует. Константы должны вводиться в форме, требуемой командой, т.е. либо в двоично-десятичном, либо в 16-ричном виде.

## 5.4 Команды фронта 0/1

Большинство команд имеют 2 версии: нормальную и версию фронта 0/1. Команды фронта 0/1 обозначаются значком @ перед мнемоникой команды.

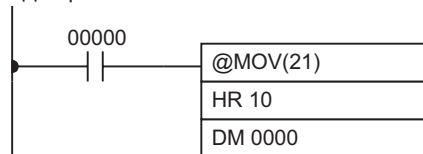
Команда “не фронта 0/1” выполняется все время, пока данная команда сканируется и условие ее выполнения = 1. Команда фронта 0/1 выполняется только один раз при изменении условия исполнения с 0 на 1. Если условие исполнения не изменялось после переключения 0/1 или изменялось с 1 на 0, команда не выполняется. Следующие 2 примера показывают работу команд MOV и @MOV, которые используются для пересылки данных по адресу, заданному первым операндом по адресу, заданному вторым операндом.

Диаграмма А



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	MOV	
		HR 10
		DM 0000

Диаграмма В



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	@MOV	
		HR 10
		DM 0000

На диаграмме А нормальная команда MOV будет пересылать содержание HR 10 в DM 0000 каждый раз, когда опрашивается 00000. Если время цикла 80 мс и 00000 остается в состоянии 1 2.0 с, операция будет выполняться 25 раз, и только последнее значение, перенесенное в DM 0000, будет там сохраняться.

На диаграмме В команда @MOV фронта 0/1 переместит содержание HR 10 в DM 0000 только один раз, когда 00000 станет = 1. Если 00000 останется = 1 в течение 2.0 с, при времени цикла 80 мс операция перемещения будет выполнена только раз при изменении 00000 из 0 в 1. Поскольку содержание HR 10 может измениться в течение 2 с, когда 00000 = 1, содержание DM 0000 после 2 с может отличаться в зависимости от того, какая команда использовалась: MOV или @MOV.

Все операнды, релейно-контактные символы и другие атрибуты команды одинаковы для обеих версий команд: нормальной и версии фронта 0/1. При вводе используются одинаковые коды команд, но вводится NOT после кода команды для указания того, что используется версия фронта 1/0. Большинство, но не все команды, имеют версию фронта 0/1.

О влиянии команд INTERLOCK и INTERLOCK CLEAR на команды фронта 0/1 см. 5.11.

В C200HX/HG/HE имеются еще две команды, срабатывающие при изменении условия исполнения: DIFU и DIFD. DIFU работает так же, как и команда фронта 0/1, но применяется для включения бита в 1 на 1 цикл. DIFD также включает бит в 1 на 1 цикл, но при изменении условия исполнения с 1 на 0. Подробности о DIFU и DIFD см. 5.9.2.



**Замечание** Не применяйте SR 25313 и SR 25315 для команд фронта 0/1. Эти биты не изменяют свое состояние и не переключают команды фронта 0/1.

## 5.5 Дополнительные команды

### 5.5.1 Дополнительные команды C200HX/HG/HE-CPU\_\_-E

C200HX/HG/HE имеет больше команд, которые требуют функционального кода (121), чем самих функциональных кодов (100), так что некоторые команды не имеют фиксированного функционального кода. Эти команды, называемые дополнительными, перечисляются в следующей таблице. Функциональные коды по умолчанию даны для команд, которым они присвоены.

Дополнительной команде можно присвоить один из 18 функциональных кодов операцией присвоения функциональных кодов с программатора. Функциональные коды - это 17, 18, 19, 47, 48, 60, 69, 87, 88 и 89. Подробности о присвоении функционального кода см. 7.2.14.

Код	Мнемоника	Наименование		Пункт
17	(@)ASFT	ASYNCHRONOUS SHIFT REGISTER	Асинхронный регистр сдвига	5.15.10
18	(@)SCAN	CYCLE TIME	Время цикла	5.25.2
19	(@)MCMP	MULTI-WORD COMPARE	Сравнение нескольких слов	5.17.1
47	(@)LMSG	32-CHACTER MESSAGE	Сообщение из 32 знаков	5.25.5
48	(@)TERM	TERMINAL MODE	Режим TERMINAL	5.25.6
60	CMPL	DOUBLE COMPARE	Сравнивает слов двойной длины	5.17.3
61	(@)MPRF	GROUP-2 HIGH-DENSITY I/O REFRESH	Обновление входов/выходов модулей группы 2	5.25.9
62	(@)XFRB	TRANSFER BIT	Передача бита	5.16.10
63	(@)LINE	COLUMN TO LINE	Преобразование столбца в строку	5.18.13
64	(@)COLM	LINE TO COLUMN	Преобразование строки в столбец	5.18.14
65	(@)SEC	HOURS TO SECOND	Часы в секунды	5.18.5
66	(@)HMS	SECONDS TO HOURS	Секунды в часы	5.18.6
67	(@)BCNT	BIT COUNTER	Счетчик битов	5.25.10
68	(@)BCMP	BLOCK COMPARE	Сравнение модуля	5.17.4
69	(@)APR	ARITHMETICAL PROCESS	Арифметический процесс	5.21.5
87	TTIM	TOTALIZING TIMER	Суммирующий таймер	5.14.3
88	ZCP	AREA RANGE COMPARE	Сравнение зоны области	5.17.6
89	(@)INT	INTERRUPT CONTROL	Управление прерываниями	5.23.6
-	7 SEG	7-SEGMENT DISPLAY OUTPUT	Выдача на 7-сегментный индикатор	5.28.1
-	(@)ADBL	Double Binary ADD	Сложение двоичных чисел двойной длины	5.20.5
-	AVG	AVERAGE VALUE	Среднее значение	5.21.3
-	(@)BXF2	EM BANK TRANSFER	Передача банка EM	5.16.12
-	CPS	SIGNED BINARY COMPARE	Сравнение двоичных чисел со знаком	5.17.8
-	CPSL	DOUBLE SIGNED BINARY COMPARE	Сравнение двоичных чисел двойной длины со знаком	5.17.9
-	(@)DBS	SIGNED BINARY DIVIDE	Деление двоичных чисел двойной длины	5.20.9
-	(@)DBSL	DOUBLE SIGNED BINARY DIVIDE	Деление двоичных чисел двойной длины со знаком	5.20.10
-	DSW	DIGITAL SWITCH INPUT	Ввод с цифрового переключателя	5.28.2
-	(@)EMBC	SELECT EM BANK	Выбор банка EM	5.25.16

## 5.5 Дополнительные команды

Код	Мнемоника	Наименование		Пункт
-	(@)FCS	FCS CALCULATE	Вычисление контрольной суммы кадра	5.25.11
-	FPD	FAILURE POINT DETECT	Обнаружение точки сбоя	5.25.12
-	(@)HEX	ASCII-TO-HEXADECIMAL	Преобразование данных ASCII в 16-ричный.	5.18.11
-	HKY	HEXADECIMAL KEY INPUT	Ввод с клавиатуры 16 клавиш.	5.28.3
-	(@)IEMS	INDIRECT EM ADDRESSING	Косвенная адресация EM	5.25.15
-	(@)IORD	SPECIAL I/O UNIT READ	Чтение модулей специальных входов/выходов	5.29.1
-	(@)IOWR	SPECIAL I/O UNIT WRITE	Запись модулей специальных входов/выходов	5.29.2
-	(@)MAX	FIND MAXIMUM	Найти максимум	5.21.1
-	(@)MBS	SIGNED BINARY MULTIPLY	Умножение двоичных чисел двойной длины	5.20.7
-	(@)MBSL	DOUBLE SIGNED BINARY MULTIPLY	Умножение двоичных чисел двойной длины со знаком	5.20.8
-	(@)MIN	FIND MINIMUM	Найти минимум	5.21.2
-	MTR	MATRIX INPUT	Ввод матрицы	5.28.5
-	(@)NEG	2'S COMPLEMENT	Дополнение до 2	5.18.15
-	(@)NEGL	DOUBLE 2'S COMPLEMENT	Дополнение до 2 чисел двойной длины	5.18.16
-	PID	PID CONTROL	ПИД-регулирование	5.21.6
-	(@)PMCR	PROTOCOL MACRO	Протокол макро	5.27.4
-	(@)RXD	RECEIVE	Прием	5.27.1
-	(@)SBBL	DOUBLE BINARY SUBTRACT	Вычитание двоичных чисел двойной длины	5.20.6
-	(@)SCL	SCALING	Масштабирование	5.18.12
-	(@)SRCH	DATA SEARCH	Поиск данных	5.25.13
-	(@)STUP	CHANGE RS-232C SETUP	Изменение параметров RS-232C	5.27.3
-	(@)SUM	SUM	Сумма	5.21.4
-	(@)TKY	TEN KEY INPUT	Ввод с клавиатуры 10 клавиш	5.28.4
-	(@)TXD	TRANSMIT	Передача	5.27.2
-	(@)XDMR	EXPANSION DM READ	Чтение расширенных DM	5.25.14
-	(@)XFR2	EM BLOCK TRANSFER	Передача модуля EM	5.16.11
-	ZCPL	DOUBLE AREA RANGE COMPARE	Сравнение зоны двойной длины	5.17.7

### 5.5.2 Дополнительные команды C200HX/HG/HE-CPU\_\_-ZE

Дополнительные команды - это группа команд, которые не имеют фиксированных функциональных кодов. Функциональные коды, зарезервированные для дополнительных команд, можно переприсваивать любым дополнительным командам по желанию. Функциональные коды по умолчанию даны для команд, которым они присвоены.

Дополнительной команде можно присвоить один из функциональных кодов операций присвоения функциональных кодов с консоли программирования. Функциональные коды - это 017, 018, 019, 047, 048, 060..069, 087, 088, 089, 114, 115, 116, 160, 161, 161, 162, 180..184, 190, 194, 195, 210..214, 222, 223, 235, 236, 237, 260, 261, 269, 280 и 281.

Подробности о присвоении функционального кода см. 7.2.14.

Подробности о присвоения функциональных кодов Инструкции по ПО.

Код	Мнемоника	Наименование		Пункт
017	(@)ASFT	ASYNCHRONOUS SHIFT REGISTER	Асинхронный регистр сдвига	5.15.10

## 5.5 Дополнительные команды

Код	Мнемоника	Наименование		Пункт
018	(@)SCAN	CYCLE TIME	Время цикла	5.25.2
019	(@)MCMP	MULTI-WORD COMPARE	Сравнение нескольких слов	5.17.1
047	(@)LMSG	32-CHACTER MESSAGE	Сообщение из 32 знаков	5.25.5
048	(@)TERM	TERMINAL MODE	Режим TERMINAL	5.25.6
060	CMPL	DOUBLE COMPARE	Сравнивает слов двойной длины	5.17.3
061	(@)MPRF	GROUP-2 HIGH-DENSITY I/O REFRESH	Обновление входов/выходов модулей группы 2	5.25.9
062	(@)XFRB	TRANSFER BIT	Передача бита	5.16.10
063	(@)LINE	COLUMN TO LINE	Преобразование столбца в строку	5.18.13
064	(@)COLM	LINE TO COLUMN	Преобразование строки в столбец	5.18.14
065	(@)SEC	HOURS TO SECOND	Часы в секунды	5.18.5
066	(@)HMS	SECONDS TO HOURS	Секунды в часы	5.18.6
067	(@)BCNT	BIT COUNTER	Счетчик битов	5.25.10
068	(@)BCMP	BLOCK COMPARE	Сравнение модуля	5.17.4
069	(@)APR	ARITHMETICAL PROCESS	Арифметический процесс	5.21.5
087	TTIM	TOTALIZING TIMER	Суммирующий таймер	5.14.3
088	ZCP	AREA RANGE COMPARE	Сравнение зоны области	5.17.6
089	(@)INT	INTERRUPT CONTROL	Управление прерываниями	5.23.6
114	CPS	SIGNED BINARY COMPARE	Сравнение двоичных чисел со знаком	5.17.8
115	CPSL	DOUBLE SIGNED BINARY COMPARE	Сравнение двоичных чисел двойной длины со знаком	5.17.9
116	ZCPL	DOUBLE AREA RANGE COMPARE	Сравнение зоны двойной длины	5.17.7
160	(@)NEG	2'S COMPLEMENT	Дополнение до 2	5.18.15
161	(@)NEGL	DOUBLE 2'S COMPLEMENT	Дополнение до 2 чисел двойной длины	5.18.16
162	(@)HEX	ASCII-TO-HEXADECIMAL	Преобразование данных ASCII в 16-ричный.	5.18.11
180	(@)FCS	FCS CALCULATE	Вычисление контрольной суммы кадра	5.25.11
181	(@)SRCH	DATA SEARCH	Поиск данных	5.25.13
182	(@)MAX	FIND MAXIMUM	Найти максимум	5.21.1
183	(@)MIN	FIND MINIMUM	Найти минимум	5.21.2
184	(@)SUM	SUM	Сумма	5.21.4
190	PID	PID CONTROL	ПИД-регулирование	5.21.6
194	(@)SCL	SCALING	Масштабирование	5.18.12
195	AVG	AVERAGE VALUE	Среднее значение	5.21.3
210	DSW	DIGITAL SWITCH INPUT	Ввод с цифрового переключателя	5.28.2
211	(@)TKY	TEN KEY INPUT	Ввод с клавиатуры 10 клавиш	5.28.4
212	HKY	HEXADECIMAL KEY INPUT	Ввод с клавиатуры 16 клавиш.	5.28.3
213	MTR	MATRIX INPUT	Ввод матрицы	5.28.5
214	7 SEG	7-SEGMENT DISPLAY OUTPUT	Выдача на 7-сегментный индикатор	5.28.1
222	(@)IORD	SPECIAL I/O UNIT READ	Чтение модулей специальных входов/выходов	5.29.1
223	(@)IOWR	SPECIAL I/O UNIT WRITE	Запись модулей специальных входов/выходов	5.29.2
235	(@)RXD	RECEIVE	Прием	5.27.1
236	(@)TXD	TRANSMIT	Передача	5.27.2

## 5.6 Кодирование выходных команд (“правосторонних”)

Код	Мнемоника	Наименование		Пункт
237	(@)STUP	CHANGE RS-232C SETUP	Изменение параметров RS-232C	5.27.3
260	(@)PMCR	PROTOCOL MACRO	Протокол макро	5.27.4
269	FPD	FAILURE POINT DETECT	Обнаружение точки сбоя	5.25.12
280	(@)XDMR	EXPANSION DM READ	Чтение расширенных DM	5.25.14
281	(@)EMBC	SELECT EM BANK	Выбор банка EM	5.25.16

Код	Мнемоника	Наименование		Пункт
-	(@)ADBL	DOUBLE BINARY ADD	Сложение двоичных чисел двойной длины	5.20.5
-	(@)SBBL	DOUBLE BINARY SUBTRACT	Вычитание двоичных чисел двойной длины	5.20.6
-	(@)MBSL	DOUBLE SIGNED BINARY MULTIPLY	Умножение двоичных чисел двойной длины со знаком	5.20.8
-	(@)DBSL	DOUBLE SIGNED BINARY DIVIDE	Деление двоичных чисел двойной длины со знаком	5.20.10
-	(@)MBS	SIGNED BINARY MULTIPLY	Умножение двоичных чисел двойной длины	5.20.7
-	(@)DBS	SIGNED BINARY DIVIDE	Деление двоичных чисел двойной длины	5.20.9
-	(@)BXF2	EM BANK TRANSFER	Передача банка EM	5.16.12
-	(@)IEMS	INDIRECT EM ADDRESSING	Косвенная адресация EM	5.25.15
-	(@)XFR2	EM BLOCK TRANSFER	Передача модуля EM	5.16.11

## 5.6 Кодирование выходных команд (“правосторонних”)

Написание мнемкода для некоторых релейно-контактных команд описано в гл. 4. Преобразование информации релейно-контактных символов для всех других команд производится по образцу, описанному ниже, и не указывается для каждой команды индивидуально.

Первое слово любой команды определяет команду и задает определители. Если команда требует только битового операнда без определителя, битовый операнд помещается в той же строке, что и мнемоника команды. Все другие операнды располагаются в строках после команды, один операнд на строку и в том же порядке, как они появляются в релейно-контактном обозначении команды.

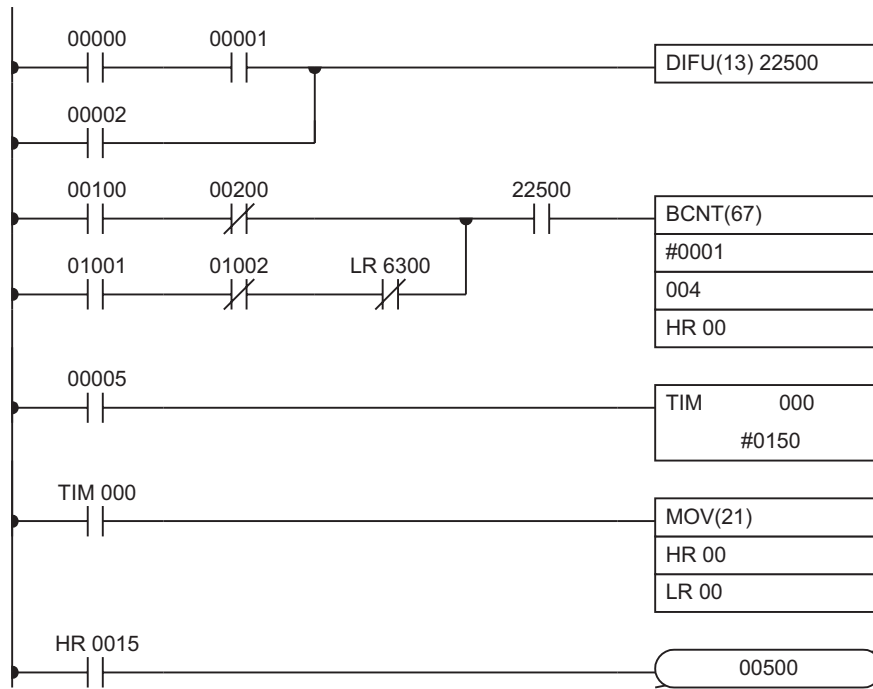
Столбцы адреса и команды таблицы мнемкодов заполняются только для слов команд. Для всех других строк два левых столбца остаются пустыми. Если команда не требует определителя или битового операнда, столбец данных остается пустым для первой строки. Рекомендуем зачеркивать все пустые места столбца данных (для всех слов команд, которые не требуют данных) чтобы быстро просмотреть столбец данных, если был пропущен какой-либо адрес.

Если в столбце данных используется адрес IR или SR, левая сторона столбца остается пустой. Если используется любая другая область данных, слева помещается сокращенное наименование области данных, а адрес располагается справа. Если вводится константа, символ # помещается в столбце данных слева, а вводимое число справа. Числа в качестве определителей к слову команды не требуют специальных символов. Биты таймеров/счетчиков, определенные как таймеры и счетчики, принимают обозначения TIM(таймер) или CNT(счетчик).

При кодировании команды, которая имеет функциональный код, обязательно пишете это функциональный код, который необходим при вводе команд с программатора. Также обязательно обозначайте команды фронта 0/1 символом @.

Следующая диаграмма и соответствующий мнемкокод иллюстрируют вышеизложенное.

## 5.6 Кодирование выходных команд (“правосторонних”)

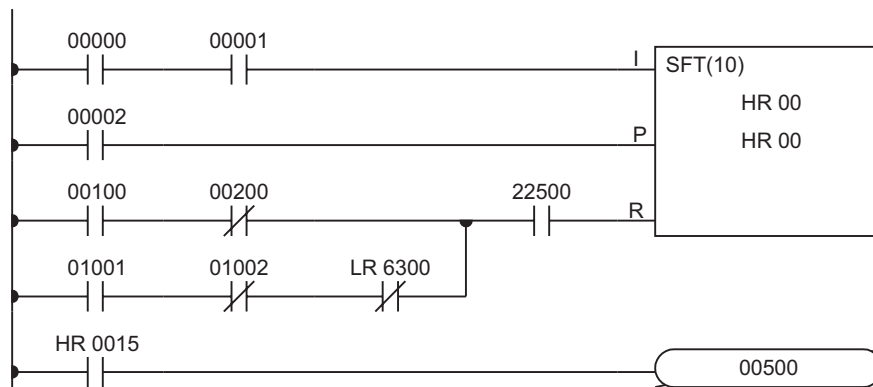


Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND	00001
00002	OR	00002
00003	DIFU	22500
00004	LD	00100
00005	AND NOT	00200
00006	LD	01001
00007	AND NOT	01002
00008	AND NOT	LR 6300
00009	OR LD	-
00010	AND	22500
00011	BCNT	
		# 0001
		004
		HR 00
00012	LD	HR 00005
00013	TIM	000
		# 0150
00014	LD	TIM 000
00015	MOV	
		HR 00
		LR 00
00016	LD	HR 00015
00017	OUT NOT	00500

### Команда с несколькими командными линиями

Если “правосторонняя” (выходная) команда требует несколько командных линий (такие, как KEEP), все командные линии команды вводятся перед самой командой. Каждая командная линия кодируется, начиная с LD или LD NOT для формирования логического

модуля, который объединяет выходная команда. Далее в качестве примера приведена команда SFT.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND	00001
00002	LD	00002
00003	LD	00100
00004	AND NOT	00200
00005	LD	01001
00006	AND NOT	01002
00007	AND NOT	LR 6300
00008	OR LD	-
00009	AND	22500
00010	SFT	
		HR 00
		HR 00
00011	LD	HR 0015
00012	OUT NOT	00500

**END**

Когда Вы закончили кодирование программы, обязательно поместите END в последний адрес.

## 5.7 Списки команд

В данной главе приводятся списки команд, существующих в C200HX/HG/HE. Первая таблица используется для нахождения команд по функциональному коду. Вторая таблица служит для нахождения команд по мнемонике. В обеих таблицах значок @ показывает вариант команд фронта 0/1.

**Замечание** Список дополнительных команд см. 5.5

### 5.7.1 Функциональные коды C200HX/HG/HE-CPU\_\_E

В следующей таблице перечислены команды CQM1, имеющие фиксированный функциональный код. Каждая команда описана мнемокодом и названием команды. Используйте цифры левого столбца в качестве старшей цифры, и цифры в верхней строке в качестве младшей цифры функционального кода.

Левый разряд	Правый разряд									
	0	1	2	3	4	5	6	7	8	9
0	NOP	END	IL	ILC	JMP	JME	(@)FAL	(@)FALS	STEP	SNXT
1	SFT	KEEP	CNTR	DIFU	DIFD	TIMH	(@)WSFT	(@)ASFT	(@)SCAN	(@)MCMP
2	CMP	(@)MOV	(@)MVN	(@)BIN	(@)BCD	(@)ASL	(@)ASR	(@)ROL	(@)ROR	(@)COM

Левый разряд	Правый разряд									
	0	1	2	3	4	5	6	7	8	9
3	(@)ADD	(@)SUB	(@)MUL	(@)DIV	(@)ANDW	(@)ORW	(@)XORW	(@)XNRW	(@)INC	(@)DEC
4	(@)STC	(@)CLC	-	-	-	TRSM	(@)MSG	(@)LMSG	(@)TERM	-
5	(@)ADB	(@)SBB	(@)MUL	(@)DVB	(@)ADDL	(@)SUBL	(@)MULL	(@)DIVL	(@)BINL	(@)BCDL
6	CMPL	(@)MPRF	(@)XFRB	(@)LINE	(@)COLM	(@)SEC	(@)HMS	(@)BCNT	(@)BCMP	(@)APR
7	(@)XFER	(@)BSET	(@)ROOT	(@)XCHG	(@)SLD	(@)SRD	(@)MLPX	(@)DMPX	(@)SDEC	(@)FDIV
8	(@)DIST	(@)COLL	(@)MOVB	(@)MOVD	(@)SFTR	(@)TCMP	(@)ASC	(@)TTIM	(@)ZCP	(@)INT
9	(@)SEND	(@)SBS	SBN	RET	(@)WDT	-	-	(@)IORF	(@)RECV	(@)MACRO

## 5.7.2 Алфавитный список мнемоники C200HX/HG/HE-CPU \_\_-E

Мнемоника	Код	Слов	Наименование		Пункт
7 SEG	-	4	7-segment display output	Выдача на 7-сегментный индикатор	5.28.1
ADB(@)	50	4	Binary ADD	Сложение двоичных чисел	5.20.1
ADBL(@)	-	4	Double Binary ADD	Сложение двоичных чисел двойной длины	5.20.5
ADD(@)	30	4	BCD ADD	Сложение двоично-десятичных чисел	5.19.5
ADDL(@)	54	4	DOUBLE BCD ADD	Сложение двоично-десятичных чисел двойной длины	5.19.6
AND	Нет	1	AND	Логическое И	5.8.1
AND LD	Нет	1	AND LOAD	ЛОГИЧЕСКОЕ И С ЗАГРУЗИТЬ	5.8.2
AND NOT	Нет	1	AND NOT	Логическое И С Инверсией	5.8.1
ANDW (@)	34	4	LOGICAL AND	Логическое И	5.22.2
APR (@)	69	4	ARITHMETIC PROCESS	Арифметические расчеты	5.21.5
ASC (@)	86	4	ASCII CONVERT	Преобразование в код ASCII	5.18.10
ASFT (@)	17	4	ASYNCHRONOUS SHIFT REGISTER	Асинхронный регистр сдвига	5.15.10
ASL (@)	25	2	ARITHMETIC SHIFT LEFT	Арифметический сдвиг влево	5.15.3
ASR (@)	26	2	ARITHMETIC SHIFT RIGHT	Арифметический сдвиг вправо	5.15.4
AVG	-	4	AVERAGE VALUE	Среднее значение	5.21.3
BCD (@)	24	3	BINARY TO BCD	Двоичное в двоично-десятичное	5.18.3
BCDL (@)	59	3	DOUBLE BINARY-TO-DOUBLE BCD	Двоичное двойной длины в двоично-десятичное двойной длины	5.18.4
BCMP (@)	68	4	BLOCK COMPARE	Сравнение модуля	5.17.4
BCNT (@)	67	4	BIT COUNTER	Битовый счетчик	5.25.10
BIN (@)	23	3	BCD-TO-BINARY	Двоично-десятичное в двоичное	5.18.1
BINL (@)	58	3	DOUBLE BCD-TO-DOUBLE BINARY	Двоично-десятичное двойной длины в двоичное двойной длины	5.18.2
BSET (@)	71	4	BLOCK SET	Заполнение модуля	5.16.3
BXF2 (@)	-	4	EM BLOCK TRANSFER	Передача модуля EM	5.16.12
CLC (@)	41	1	CLEAR CARRY	Очистка переноса	5.19.4
CMP	20	3	COMPARE	Сравнить	5.17.2
CMPL	60	4	DOUBLE COMPARE	Сравнить числа двойной длины	5.17.3
CNT	Нет	2	COUNTER	Счетчик	5.14.4

## 5.7 Списки команд

Мнемоника	Код	Слов	Наименование		Пункт
CNTR	12	3	REVERSIBLE COUNTER	Реверсивный счетчик	5.14.5
COLL (@)	81	4	DATA COLLECT	Собрать данные	5.16.7
COLM (@)	64	4	LINE TO COLUMN	Строку в столбец	5.18.14
COM (@)	29	2	COMPLEMENT	Дополнение	5.22.1
CPS	-	4	SIGNED BINARY COMPARE	Сравнение двоичных чисел со знаком	5.17.8
CPSL	-	4	DOUBLE SIGNED BINARY COMPARE	Сравнение двоичных чисел двойной длины со знаком	5.17.9
DBS (@)	-	4	SIGNED BINARY DIVIDE	Деление двоичных чисел со знаком	5.20.9
DBSL (@)	-	4	DOUBLE SIGNED BINARY DIVIDE	Деление двоичных чисел двойной длины со знаком	5.20.10
DEC (@)	39	2	BCD DECREMENT	Декремент двоично-десятичного числа	5.19.2
DIFD	14	2	DIFFERENTIATE DOUN	Включение на 1 цикл по фронту 1/0	5.9.2
DIFU	13	2	DIFFERENTIATE UP	Включение на 1 цикл по фронту 0/1	5.9.2
DIST (@)	80	4	SINGLE WORD DISTRIBUTE	Распределение одного слова	5.16.6
DIV (@)	33	4	BCD DIVIDE	Деление двоично-десятичного числа	5.19.11
DIVL (@)	57	4	DOUBLE BCD DIVIDE	Деление двоично-десятичного числа двойной длины	5.19.12
DMPX (@)	77	4	16-TO-4 ENCODER	Преобразование 16-в-4	5.18.8
DSW	-	4	DIGITAL SWITCH	Ввод с кодового колеса	5.28.2
DVB (@)	53	4	BINARY DIVIDE	Деление двоичных чисел	5.20.4
EMBC (@)	-	2	SELECT EM BANK	Выбрать банк EM	5.25.16
END	01	1	END	Конец программы	5.12.
FAL (@)	06	2	FAILURE ALARM AND RESET	Признак ошибки и сброс	5.25.1
FALS	07	2	SEVERE FAILURE ALARM	Признак фатальной ошибки	5.25.1
FCS (@)	-	4	FCS CALCULATE	Подсчет КСК (контрольной суммы конверта)	5.25.11
FDIV (@)	79	4	FLOATING POINT DIVIDE	Деление чисел с плавающей точкой	5.19.13
FPD	-	4	FAILURE POINT DETECT	Поиск точки сбоя	5.25.12
HEX (@)	-	4	ASCII-TO-HEXADECIMAL	ASCII в 16-ричный вид	5.18.11
HKY	-	4	HEXADECIMAL KEY INPUT	Ввод с клавиатуры 16 клавиш	5.28.3
HMS (@)	66	4	SECOND TO HOURS	Секунды в часы	5.18.6
IEMS (@)	-	2	INDIRECT EM ADDRESS	Косвенная адресация EM	5.25.15
IL	02	1	INTERLOCK	Секция INTERLOCK	5.10
ILC	03	1	INTERLOCK CLEAR	Конец секции INTERLOCK	5.10
INC (@)	38	2	INCREMENT	Инкремент	5.19.1
INT (@)	89	4	INTERRUPT CONTROL	Управление прерываниями	5.23.6
IORD (@)	-	4	SPECIAL I/O UNIT READ	Чтение специального модуля входов/выходов	5.25.8
IORF (@)	97	3	I/O REFRESH	Обновление входов/выходов	5.29.1
IOWR (@)	-	4	SPECIAL I/O UNIT WRITE	Запись специального модуля входов/выходов	5.29.2
JME	05	2	JUMP END	Конец перехода	5.11



## 5.7 Списки команд

Мнемоника	Код	Слов	Наименование		Пункт
JMP	04	2	JUMP	Переход	5.11
KEEP	11	2	KEEP	Сохранить значение	5.9.4
LD	Нет	1	LOAD	Загрузка	5.8.1
LD NOT	Нет	1	LOAD NOT	Загрузка инверсии	5.8.1
LINE (@)	63	4	COLUMN-TO-LINE	Столбец в строку	5.18.13
LMSG (@)	47	4	32-CHARACTER MESSAGE	Сообщение 32 знака	5.25.5
MAX (@)	-	4	FIND MAXIMUM	Найти максимум	5.21.1
MBS (@)	-	4	SIGNED BINARY MULTIPLY	Умножение двоичных чисел со знаком	5.20.7
MBSL (@)	-	4	DOUBLE SIGNED BINARY MULTIPLY	Умножение двоичных чисел двойной длины со знаком	5.20.8
MCMP (@)	19	4	MULTI-WORD COMPARE	Сравнение нескольких слов	5.17.1
MCRO (@)	99	4	MACRO	Макрокоманда	5.23.5
MIN (@)	-	4	FIND MINIMUM	Найти минимум	5.21.2
MLB (@)	52	4	BINARY MULTIPLY	Умножение двоичных чисел	5.20.3
MLPX (@)	76	4	4-TO-16 DECODER	Преобразование 4-в-16	5.18.7
MOV (@)	21	3	MOVE	Пересылка	5.16.1
MOVB (@)	82	4	MOVE BIT	Пересылка бита	5.16.8
MOVD (@)	83	4	MOVE DIGIT	Пересылка слова	5.16.9
MPRF (@)	61	4	GROUP-2 HIGH-DENSITY I/O REFRESH	Обновление входов/выходов	5.25.9
MSG (@)	46	2	MESSAGE	Сообщение	5.25.4
MTR	-	4	MATRIX INPUT	Ввод матрицы	5.28.5
MUL (@)	32	4	BCD MULTIPLY	Умножение двоично-десятичных чисел	5.19.9
MULL (@)	56	4	DOUBLE BCD MULTIPLY	Умножение двоично-десятичных чисел двойной длины	5.19.10
MVN (@)	22	3	MOVE NOT	Пересылка инверсии	5.16.2
NEG (@)	-	4	2'S COMPLEMENT	Дополнение до 2	5.18.15
NEGL (@)	-	4	DOUBLE 2'S COMPLEMENT	Дополнение до 2 чисел двойной длины	5.18.16
NOP	00	1	NO OPERATION	Нет операции	5.13
OR	Нет	1	OR	Логическое ИЛИ	5.8.1
OR LD	Нет	1	OR LOAD	Логическое Или С Загрузить	5.8.2
OR NOT	Нет	1	OR NOT	Логическое Или С Инверсией	5.8.1
ORW (@)	35	4	LOGICAL OR	Логическое Или Двух Слов	5.22.3
OUT	Нет	2	OUTPUT	Выход	5.9.1
OUT NOT	Нет	2	OUTPUT NOT	Выход инверсии	5.9.1
PID(@)	-	4	PID CONTROL	ПИД-регулятор (пропорционально-интегральный)	5.21.6
PMCR (@)	-	4	PROTOCOL MACRO	PROTOCOL MACRO	5.27.4
RECV (@)	98	4	NETWORK RECEIVE	Прием из сети	5.26.2
RET	93	1	SUBROUTINE RETURN	Возврат из подпрограммы	5.23.4
ROL (@)	27	2	ROTATE LEFT	Циклический сдвиг влево	5.15.5
ROOT (@)	72	3	SQUARE ROOT	Квадратный корень	5.19.14
ROR (@)	28	2	ROTATE RIGHT	Циклический сдвиг вправо	5.15.6
RSET	Нет	2	RESET	Сброс	5.9.3
RXD (@)	-	4	RECEIVE	Прием	5.27.1

## 5.7 Списки команд

Мнемоника	Код	Слов	Наименование		Пункт
SBB (@)	51	4	BINARY SUBTRACT	Вычитание двоичных чисел	5.20.2
SBBL (@)	-	4	DOUBLE BINARY SUBTRACT	Вычитание двоичных чисел двойной длины	5.20.6
SBN	92	2	SUBROUTINE DEFINE	Определить подпрограмму	5.23.4
SBS (@)	91	2	SUBROUTINE ENTER	Вызвать подпрограмму	5.23.3
SCAN (@)	18	4	CYCLE TIME	Время цикла	5.25.2
SCL (@)	-	4	SCALING	Масштабирование	5.18.12
SDEC (@)	78	4	7-SEGMENT DECODER	Декодирование 7-сегментного индикатора.	5.18.9
SEC	65	4	HOURS TO SECONDS	Часы в секунды	5.18.5
SEND (@)	90	4	NETWORK SEND	Посылка в сеть	5.26.1
SET	Нет	2	SET	Установить в 1	5.9.3
SFT	10	3	SHIFT REGISTER	Регистр сдвига	5.15.1
SFTR (@)	84	4	REVERSIBLE SHIFT REGISTER	Реверсивный регистр сдвига	5.15.2
SLD (@)	74	3	ONE DIGIT SHIFT LEFT	Сдвиг одной цифры влево	5.15.7
SNXT	09	2	STEP START	Пуск секции STEP	5.24.1
SRCH (@)	-	4	DATA SEARCH	Поиск данных	5.25.13
SRD (@)	75	3	ONE DIGIT SHIFT RIGHT	Сдвиг одной цифры вправо	5.15.8
STC (@)	40	1	SET CARRY	Установить перенос	5.19.3
STEP	08	2	STEP DEFINE	Определить секцию STEP	5.24.1
STUP (@)	-	2	CHANGE RS-232C SETUP	Изменить параметры RS-232C	5.27.3
SUB (@)	31	4	BCD SUBTRACT	Вычитание двоичных чисел	5.19.7
SUBL (@)	55	4	DOUBLE BCD SUBTRACT	Вычитание двоичных чисел двойной длины	5.19.8
SUM (@)	-	4	SUM	Сумма	5.21.4
TCMP (@)	85	4	TABLE COMPARE	Сравнение значение с таблицей	5.17.5
TERM (@)	48	4	TERMINAL MODE	Режим терминала	5.25.6
TIM	Нет	2	TIMER	Таймер	5.14.1
TIMH	15	3	HIGH-SPEED TIMER	Высокоскоростной таймер	5.14.2
TKY (@)	-	4	TEN KEY INPUT	Ввод с клавиатуры 10 клавиш	5.28.4
TRSM	45	1	TRACE MEMORY SAMPLE	Выборка для трассировки памяти	5.25.3
TTIM	87	4	TOTALIZING TIMER	Суммирующий таймер	5.14.3
TXD (@)	-	4	TRANSMIT	Передача	5.27.2
WDT (@)	94	2	WATCHDOG TIMER REFRESH	Обновление синхронизирующего таймера	5.25.7
WSFT (@)	16	3	WORD SHIFT	Сдвиг слова	5.15.9
XCHG (@)	73	3	DATA EXCHANGE	Обмен данными	5.16.5
XDMR (@)	-	4	EXPANSION DM READ	Чтение расширенных DM	5.25.14
XFER (@)	70	4	BLOCK TRANSFER	Передача модуля	5.16.4
XFR2 (@)	-	4	EM BLOCK TRANSFER	Передача модуля EM	5.16.11
XFRB (@)	62	4	TRANSFER BITS	Передача битов	5.16.10
XNRW (@)	37	4	EXCLUSIVE NOR	Исключая ИЛИ-НЕ	5.22.5
XORW (@)	36	4	EXCLUSIVE OR	ИСКЛЮЧАЯ ИЛИ	5.22.4
ZCP	88	4	AREA RANGE COMPARE	Сравнение с зоной	5.17.6

Мнемоника	Код	Слов	Наименование	Пункт
ZCPL	-	4	DOUBLE AREA RANGE COMPARE	Сравнение с зоной чисел двойной длины 5.17.7

### 5.7.3 Функциональные коды C200HX/HG/HE-CPU\_\_-ZE

В следующей таблице перечислены команды CQM1, имеющие фиксированный функциональный код и дополнительные инструкции, имеющие функциональный код по умолчанию. Каждая команда описана мнемокодом и названием команды. Используйте цифры левого столбца в качестве старшей цифры, и цифры в верхней строке в качестве младшей цифры функционального кода.

Левый разряд	Правый разряд									
	0	1	2	3	4	5	6	7	8	9
00	NOP	END	IL	ILC	JMP	JME	(@)FAL	(@)FALS	STEP	SNXT
01	SFT	KEEP	CNTR	DIFU	DIFD	TIMH	(@)WSFT	(@)ASFT	(@)SCAN	(@)MCMP
02	CMP	(@)MOV	(@)MVN	(@)BIN	(@)BCD	(@)ASL	(@)ASR	(@)ROL	(@)ROR	(@)COM
03	(@)ADD	(@)SUB	(@)MUL	(@)DIV	(@)ANDW	(@)ORW	(@)XORW	(@)XNRW	(@)INC	(@)DEC
04	(@)STC	(@)CLC	-	-	-	TRSM	(@)MSG	(@)LMSG	(@)TERM	-
05	(@)ADB	(@)SBB	(@)MBL	(@)DVB	(@)ADDL	(@)SUBL	(@)MULL	(@)DIVL	(@)BINL	(@)BCDL
06	CMPL	(@)MPRF	(@)XFRB	(@)LINE	(@)COLM	(@)SEC	(@)HMS	(@)BCNT	(@)BCMP	(@)APR
07	(@)XFER	(@)BSET	(@)ROOT	(@)XCHG	(@)SLD	(@)SRD	(@)MLPX	(@)DMPX	(@)SDEC	(@)FDIV
08	(@)DIST	(@)COLL	(@)MOVB	(@)MOVD	(@)SFTR	(@)TCMP	(@)ASC	(@)TTIM	(@)ZCP	(@)INT
09	(@)SEND	(@)SBS	SBN	RET	(@)WDT	-	-	(@)IORF	(@)RECV	(@)MACRO
11	-	-	-	-	(@)CPS	(@)CPSL	(@)ZCPSL	-	-	-
16	(@)NEG	(@)NEGL	(@)HEX	-	-	-	-	-	-	-
18	(@)FCS	(@)SRCH	(@)MAX	(@)MIN	(@)SUM	-	-	-	-	-
19	PID	-	-	-	(@)SCL	(@)AVG	-	-	-	-
21	DSW	TKY	HKY	MTR	7SEG	-	-	-	-	-
22	-	-	(@)IORD	(@)IOWR	-	-	-	-	-	-
23	-	-	-	-	-	(@)RXD	(@)TXD	(@)STUP	-	-
26	(@)PMCR	(@)CMCR	-	-	-	-	-	-	-	FPD
28	(@)XDMR	(@)EMBC	-	-	-	-	-	-	-	-
30										
31										
32										
35	TST	TSTN	-	-	-	-	-	-	-	-
40	(@)+	(@)+L	(@)+C	(@)+CL	(@)+B	(@)+BL	(@)+BC	(@)+BCL	-	-
41	(@)-	(@)-L	(@)-C	(@)-CL	(@)-B	(@)-BL	(@)-BC	(@)-BCL	-	-
42	(@)*	(@)*L	(@)*U	(@)*UL	(@)*B	(@)*BL	-	-	-	-
43	(@)/	(@)/L	(@)/U	(@)/UL	(@)/B	(@)/BL	-	-	-	-
48	(@)ADBL	(@)SDBL	(@)MBSL	(@)DBSL	(@)MBS	(@)DBS	-	-	-	-

### 5.7.4 Алфавитный список мнемоники C200HX/HG/HE-CPU\_\_-ZE

В следующей таблице перечислены команды C200HX/HG/HE-CPU\_\_-ZE в алфавитном порядке. Функциональные коды по умолчанию для дополнительных команд даны в скобках. Код (-) показан для дополнительных команд, у которых нет функциональных кодов по умолчанию.)

Мнемоника	Код	Слов	Наименование	Пункт
7 SEG	(214)	4	7-segment display output	Выдача на 7-сегментный индикатор 5.28.1
ADB(@)	050	4	Binary ADD	Сложение двоичных чисел 5.20.1
ADBL(@)	(-)	4	Double Binary ADD	Сложение двоичных чисел двойной длины 5.20.5
ADD(@)	030	4	BCD ADD	Сложение двоично-десятичных чисел 5.19.5
ADDL(@)	054	4	DOUBLE BCD ADD	Сложение двоично-десятичных чисел двойной длины 5.19.6

## 5.7 Списки команд

Мнемоника	Код	Слов	Наименование		Пункт
AND	Нет	1	AND	Логическое И	5.8.1
AND LD	Нет	1	AND LOAD	ЛОГИЧЕСКОЕ И С ЗАГРУЗИТЬ	5.8.2
AND NOT	Нет	1	AND NOT	Логическое И С Инверсией	5.8.1
ANDW (@)	034	4	LOGICAL AND	Логическое И	5.22.2
APR (@)	(069)	4	ARITHMETIC PROCESS	Арифметические расчеты	5.21.5
ASC (@)	086	4	ASCII CONVERT	Преобразование в код ASCII	5.18.10
ASFT (@)	(017)	4	ASYNCHRONOUS SHIFT REGISTER	Асинхронный регистр сдвига	5.15.10
ASL (@)	025	2	ARITHMETIC SHIFT LEFT	Арифметический сдвиг влево	5.15.3
ASR (@)	026	2	ARITHMETIC SHIFT RIGHT	Арифметический сдвиг вправо	5.15.4
AVG	(195)	4	AVERAGE VALUE	Среднее значение	5.21.3
BCD (@)	024	3	BINARY TO BCD	Двоичное в двоично-десятичное	5.18.3
BCDL (@)	059	3	DOUBLE BINARY-TO-DOUBLE BCD	Двоичное двойной длины в двоично-десятичное двойной длины	5.18.4
BCMP (@)	(068)	4	BLOCK COMPARE	Сравнение модуля	5.17.4
BCNT (@)	(067)	4	BIT COUNTER	Битовый счетчик	5.25.10
BIN (@)	023	3	BCD-TO-BINARY	Двоично-десятичное в двоичное	5.18.1
BINL (@)	058	3	DOUBLE BCD-TO-DOUBLE BINARY	Двоично-десятичное двойной длины в двоичное двойной длины	5.18.2
BSET (@)	071	4	BLOCK SET	Заполнение модуля	5.16.3
BXF2 (@)	(-)	4	EM BLOCK TRANSFER	Передача модуля EM	5.16.12
CLC (@)	041	1	CLEAR CARRY	Очистка переноса	5.19.4
CMCR (@)	261	5	PCMCIA CARD MACRO		
CMP	020	3	COMPARE	Сравнить	5.17.2
CMPL	(060)	4	DOUBLE COMPARE	Сравнить числа двойной длины	5.17.3
CNT	Нет	2	COUNTER	Счетчик	5.14.4
CNTR	(012)	3	REVERSIBLE COUNTER	Реверсивный счетчик	5.14.5
COLL (@)	(081)	4	DATA COLLECT	Собрать данные	5.16.7
COLM (@)	(064)	4	LINE TO COLUMN	Строку в столбец	5.18.14
COM (@)	029	2	COMPLEMENT	Дополнение	5.22.1
CPS	(114)	4	SIGNED BINARY COMPARE	Сравнение двоичных чисел со знаком	5.17.8
CPSL	(115)	4	DOUBLE SIGNED BINARY COMPARE	Сравнение двоичных чисел двойной длины со знаком	5.17.9
DBS (@)	(-)	4	SIGNED BINARY DIVIDE	Деление двоичных чисел со знаком	5.20.9
DBSL (@)	(-)	4	DOUBLE SIGNED BINARY DIVIDE	Деление двоичных чисел двойной длины со знаком	5.20.10
DEC (@)	039	2	BCD DECREMENT	Декремент двоично-десятичного числа	5.19.2
DIFD	014	2	DIFFERENTIATE DOWN	Включение на 1 цикл по фронту 1/0	5.9.2
DIFU	013	2	DIFFERENTIATE UP	Включение на 1 цикл по фронту 0/1	5.9.2
DIST (@)	080	4	SINGLE WORD DISTRIBUTE	Распределение одного слова	5.16.6

## 5.7 Списки команд

Мнемоника	Код	Слов	Наименование		Пункт
DIV (@)	033	4	BCD DIVIDE	Деление двоично-десятичного числа	5.19.11
DIVL (@)	057	4	DOUBLE BCD DIVIDE	Деление двоично-десятичного числа двойной длины	5.19.12
DMPX (@)	077	4	16-TO-4 ENCODER	Преобразование 16-в-4	5.18.8
DSW	(210)	4	DIGITAL SWITCH	Ввод с кодового колеса	5.28.2
DVB (@)	053	4	BINARY DIVIDE	Деление двоичных чисел	5.20.4
EMBC (@)	281	2	SELECT EM BANK	Выбрать банк EM	5.25.16
END	001	1	END	Конец программы	5.12.
FAL (@)	006	2	FAILURE ALARM AND RESET	Признак ошибки и сброс	5.25.1
FALS	007	2	SEVERE FAILURE ALARM	Признак фатальной ошибки	5.25.1
FCS (@)	(180)	4	FCS CALCULATE	Подсчет КСК (контрольной суммы конверта)	5.25.11
FDIV (@)	079	4	FLOATING POINT DIVIDE	Деление чисел с плавающей точкой	5.19.13
FPD	(269)	4	FAILURE POINT DETECT	Поиск точки сбоя	5.25.12
HEX (@)	162	4	ASCII-TO-HEXADECIMAL	ASCII в 16-ричный вид	5.18.11
HKY	(212)	4	HEXADECIMAL KEY INPUT	Ввод с клавиатуры 16 клавиш	5.28.3
HMS (@)	066	4	SECOND TO HOURS	Секунды в часы	5.18.6
IEMS (@)	(-)	2	INDIRECT EM ADDRESS	Косвенная адресация EM	5.25.15
IL	002	1	INTERLOCK	Секция INTERLOCK	5.10
ILC	003	1	INTERLOCK CLEAR	Конец секции INTERLOCK	5.10
INC (@)	038	2	INCREMENT	Инкремент	5.19.1
INT (@)	(089)	4	INTERRUPT CONTROL	Управление прерываниями	5.23.6
IORD (@)	222	4	SPECIAL I/O UNIT READ	Чтение специального модуля входов/выходов	5.25.8
IORF (@)	097	3	I/O REFRESH	Обновление входов/выходов	5.29.1
IOWR (@)	223	4	SPECIAL I/O UNIT WRITE	Запись специального модуля входов/выходов	5.29.2
JME	005	2	JUMP END	Конец перехода	5.11
JMP	004	2	JUMP	Переход	5.11
KEEP	011	2	KEEP	Сохранить значение	5.9.4
LD	Нет	1	LOAD	Загрузка	5.8.1
LD NOT	Нет	1	LOAD NOT	Загрузка инверсии	5.8.1
LINE (@)	(063)	4	COLUMN-TO-LINE	Столбец в строку	5.18.13
LMSG (@)	(047)	4	32-CHARACTER MESSAGE	Сообщение 32 знака	5.25.5
MAX (@)	(182)	4	FIND MAXIMUM	Найти максимум	5.21.1
MBS (@)	(-)	4	SIGNED BINARY MULTIPLY	Умножение двоичных чисел со знаком	5.20.7
MBSL (@)	(-)	4	DOUBLE SIGNED BINARY MULTIPLY	Умножение двоичных чисел двойной длины со знаком	5.20.8
MCMP (@)	(019)	4	MULTI-WORD COMPARE	Сравнение нескольких слов	5.17.1
MCRO (@)	099	4	MACRO	Макрокоманда	5.23.5
MIN (@)	(183)	4	FIND MINIMUM	Найти минимум	5.21.2
MLB (@)	052	4	BINARY MULTIPLY	Умножение двоичных чисел	5.20.3
MLPX (@)	076	4	4-TO-16 DECODER	Преобразование 4-в-16	5.18.7
MOV (@)	021	3	MOVE	Пересылка	5.16.1

## 5.7 Списки команд

Мнемоника	Код	Слов	Наименование		Пункт
MOVB (@)	082	4	MOVE BIT	Пересылка бита	5.16.8
MOVD (@)	083	4	MOVE DIGIT	Пересылка слова	5.16.9
MPRF (@)	(061)	4	GROUP-2 HIGH-DENSITY I/O REFRESH	Обновление входов/выходов	5.25.9
MSG (@)	046	2	MESSAGE	Сообщение	5.25.4
MTR	(213)	4	MATRIX INPUT	Ввод матрицы	5.28.5
MUL (@)	032	4	BCD MULTIPLY	Умножение двоично-десятичных чисел	5.19.9
MULL (@)	056	4	DOUBLE BCD MULTIPLY	Умножение двоично-десятичных чисел двойной длины	5.19.10
MVN (@)	022	3	MOVE NOT	Пересылка инверсии	5.16.2
NEG (@)	(160)	4	2'S COMPLEMENT	Дополнение до 2	5.18.15
NEGL (@)	(161)	4	DOUBLE 2'S COMPLEMENT	Дополнение до 2 чисел двойной длины	5.18.16
NOP	000	1	NO OPERATION	Нет операции	5.13
OR	Нет	1	OR	Логическое ИЛИ	5.8.1
OR LD	Нет	1	OR LOAD	Логическое Или С Загрузить	5.8.2
OR NOT	Нет	1	OR NOT	Логическое Или С Инверсией	5.8.1
ORW (@)	035	4	LOGICAL OR	Логическое Или Двух Слов	5.22.3
OUT	Нет	2	OUTPUT	Выход	5.9.1
OUT NOT	Нет	2	OUTPUT NOT	Выход инверсии	5.9.1
PID(@)	(190)	4	PID CONTROL	ПИД-регулятор (пропорционально-интегральный)	5.21.6
PMCR (@)	260	4	PROTOCOL MACRO	PROTOCOL MACRO	5.27.4
RECV (@)	098	4	NETWORK RECEIVE	Прием из сети	5.26.2
RET	093	1	SUBROUTINE RETURN	Возврат из подпрограммы	5.23.4
ROL (@)	027	2	ROTATE LEFT	Циклический сдвиг влево	5.15.5
ROOT (@)	072	3	SQUARE ROOT	Квадратный корень	5.19.14
ROR (@)	028	2	ROTATE RIGHT	Циклический сдвиг вправо	5.15.6
RSET	Нет	2	RESET	Сброс	5.9.3
RXD (@)	(235)	4	RECEIVE	Прием	5.27.1
SBB (@)	051	4	BINARY SUBTRACT	Вычитание двоичных чисел	5.20.2
SBBL (@)	(-)	4	DOUBLE BINARY SUBTRACT	Вычитание двоичных чисел двойной длины	5.20.6
SBN	092	2	SUBROUTINE DEFINE	Определить подпрограмму	5.23.4
SBS (@)	091	2	SUBROUTINE ENTER	Вызвать подпрограмму	5.23.3
SCAN (@)	(018)	4	CYCLE TIME	Время цикла	5.25.2
SCL (@)	(194)	4	SCALING	Масштабирование	5.18.12
SDEC (@)	078	4	7-SEGMENT DECODER	Декодирование 7-сегментного индикатора.	5.18.9
SEC	(065)	4	HOURS TO SECONDS	Часы в секунды	5.18.5
SEND (@)	090	4	NETWORK SEND	Посылка в сеть	5.26.1
SET	Нет	2	SET	Установить в 1	5.9.3
SFT	010	3	SHIFT REGISTER	Регистр сдвига	5.15.1
SFTR (@)	084	4	REVERSIBLE SHIFT REGISTER	Реверсивный регистр сдвига	5.15.2
SLD (@)	074	3	ONE DIGIT SHIFT LEFT	Сдвиг одной цифры влево	5.15.7
SNXT	009	2	STEP START	Пуск секции STEP	5.24.1

## 5.7 Списки команд

Мнемоника	Код	Слов	Наименование		Пункт
SRCH (@)	(181)	4	DATA SEARCH	Поиск данных	5.25.13
SRD (@)	075	3	ONE DIGIT SHIFT RIGHT	Сдвиг одной цифры вправо	5.15.8
STC (@)	040	1	SET CARRY	Установить перенос	5.19.3
STEP	008	2	STEP DEFINE	Определить секцию STEP	5.24.1
STUP (@)	237	2	CHANGE RS-232C SETUP	Изменить параметры RS-232C	5.27.3
SUB (@)	031	4	BCD SUBTRACT	Вычитание двоичных чисел	5.19.7
SUBL (@)	055	4	DOUBLE BCD SUBTRACT	Вычитание двоичных чисел двойной длины	5.19.8
SUM (@)	(184)	4	SUM	Сумма	5.21.4
TCMP (@)	085	4	TABLE COMPARE	Сравнение значение с таблицей	5.17.5
TERM (@)	(048)	4	TERMINAL MODE	Режим терминала	5.25.6
TIM	Нет	2	TIMER	Таймер	5.14.1
TIMH	015	3	HIGH-SPEED TIMER	Высокоскоростной таймер	5.14.2
TKY	(211)	4	TEN KEY INPUT	Ввод с клавиатуры 10 клавиш	5.28.4
TRSM	045	1	TRACE MEMORY SAMPLE	Выборка для трассировки памяти	5.25.3
TST	350	5	BIT TEST	Проверка бита	5.9.5
TSTN	351	5	BIT TEST NOT	Проверка инверсного бита	5.9.5
TTIM	087	4	TOTALIZING TIMER	Суммирующий таймер	5.14.3
TXD (@)	(236)	4	TRANSMIT	Передача	5.27.2
WDT (@)	094	2	WATCHDOG TIMER REFRESH	Обновление синхронизирующего таймера	5.25.7
WSFT (@)	016	3	WORD SHIFT	Сдвиг слова	5.15.9
XCHG (@)	073	3	DATA EXCHANGE	Обмен данными	5.16.5
XDMR (@)	(280)	4	EXPANSION DM READ	Чтение расширенных DM	5.25.14
XFER (@)	070	4	BLOCK TRANSFER	Передача модуля	5.16.4
XFR2 (@)	(-)	4	EM BLOCK TRANSFER	Передача модуля EM	5.16.11
XFRB (@)	(062)	4	TRANSFER BITS	Передача битов	5.16.10
XNRW (@)	037	4	EXCLUSIVE NOR	Исключая ИЛИ-НЕ	5.22.5
XORW (@)	036	4	EXCLUSIVE OR	ИСКЛЮЧАЯ ИЛИ	5.22.4
ZCP	(088)	4	AREA RANGE COMPARE	Сравнение с зоной	5.17.6
ZCPL	(116)	4	DOUBLE AREA RANGE COMPARE	Сравнение с зоной чисел двойной длины	5.17.7

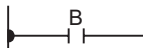
## 5.8 Команды релейно-контактной схемы

Команды релейно-контактной схемы включают команды опроса условия и блоковые команды и соответствуют условиям на релейно-контактной схеме. Блоковые команды используются для связи более сложных участков.

### 5.8.1 LOAD, LOAD NOT, AND, AND NOT, OR и OR NOT

#### LOAD - LD

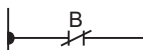
##### Обозначение на РКС



Номер инструкции		Операнды		
Контроллер	Nr	B		
		бит		
C200H□-CPU□□-E	-	IR, SR, TR, HR, AR, LR, TC		
C200H□-CPU□□-ZE	-	IR, SR, TR, HR, AR, LR, TC		

#### LOAD NOT - LD NOT

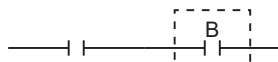
##### Обозначение на РКС



Номер инструкции		Операнды		
Контроллер	Nr	B		
		бит		
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC		
C200H□-CPU□□-ZE	-	IR, SR, HR, AR, LR, TC		

#### AND - AND

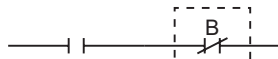
##### Обозначение на РКС



Номер инструкции		Операнды		
Контроллер	Nr	B		
		бит		
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC		
C200H□-CPU□□-ZE	-	IR, SR, HR, AR, LR, TC		

#### AND NOT - AND NOT

##### Обозначение на РКС



Номер инструкции		Операнды		
Контроллер	Nr	B		
		бит		
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC		
C200H□-CPU□□-ZE	-	IR, SR, HR, AR, LR, TC		

#### OR - OR

##### Обозначение на РКС



Номер инструкции		Операнды		
Контроллер	Nr	B		
		бит		
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC		
C200H□-CPU□□-ZE	-	IR, SR, HR, AR, LR, TC		



**OR NOT - OR NOT****Обозначение на РКС**

Номер инструкции		Операнды		
Контроллер	№	B		
C200H□-CPU□□-E	-	бит		
C200H□-CPU□□-ZE	-	IR, SR, HR, AR, LR, TC		

**Ограничения**

Ограничений количества любой из этих команд или порядка их применения нет, если программа вмещается в отведенную память.

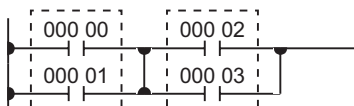
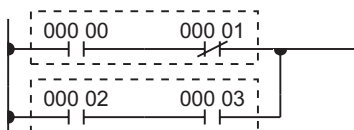
**Описание**

Данные 6 базовых команд соответствуют условиям на релейно-контактной схеме. Как описано в гл. 4, состояние битов, привязанных каждой из этих команд, определяет условие исполнения дальнейших команд. Каждая из этих команд и адрес бита может использоваться столько раз, сколько необходимо. Каждый бит может использоваться в стольких командах, в скольких требуется.

Состояние битового операнда (B), присвоенного командам LD и LD NOT, определяет первое условие исполнения. AND выполняет логическое И со своим условием исполнения и состоянием своего битового операнда; AND NOT выполняет логическое И с условием исполнения и инверсией своего битового операнда. OR выполняет логическое ИЛИ с условием исполнения и состоянием своего битового операнда; OR NOT выполняет логическое ИЛИ с условием исполнения и инверсией своего битового операнда. Символ для битов TR отличается от вышеназванных. Подробности см. 4.4.3.

**Флаги**

Данные команды не оказывают воздействия на флаги

**5.8.2 AND LOAD и OR LOAD****AND LOAD - AND LD****OR LOAD - OR LD****Описание**

Когда команды объединены в модули, которые нельзя запрограммировать только командами AND и OR, применяются AND LD и OR LD. В то время, как AND и OR используются для логического объединения состояния битов и условий исполнения, AND LD и OR LD логически объединяют два условия исполнения, текущего и последнего неиспользованного.

Для написания релейно-контактной схемы нет необходимости применять AND LD и OR LD, не нужны они и при прямом вводе релейно-контактной схемы с LSS. Однако они требуются при преобразовании программы в мнемонику и вводе в данной форме. Процедуры, ограничения процедур и примеры приведены в 4.7.

Для сокращения числа команд, требующихся при программировании, требуется понимание блоковых команд. Введение в логические модули см. 4.4.6.

**Флаги**

Данные команды не оказывают воздействия на флаги

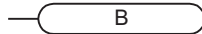
## 5.9 Команды управления битами

Есть 5 команд, которые используются для управления состоянием индивидуального бита. Это OUT, OUT NOT, DIFU, DIFD и KEEP. Данные команды используются для переключения бита в состояние 1 или 0 различными методами.

### 5.9.1 OUT и OUT NOT - Вывод и Вывод инверсии

#### OUTPUT - OUT

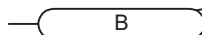
##### Обозначение на РКС



Номер инструкции		Операнды		
Контроллер	№	В		
		бит		
C200H□-CPU□□-E	-	IR, SR, TR, HR, AR, LR, TC		
C200H□-CPU□□-ZE	-	IR, SR, TR, HR, AR, LR, TC		

#### OUTPUT NOT - OUT NOT

##### Обозначение на РКС



Номер инструкции		Операнды		
Контроллер	№	В		
		бит		
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC		
C200H□-CPU□□-ZE	-	IR, SR, HR, AR, LR, TC		

#### Ограничения

Любой выходной бит можно использовать только с одной командой, которая управляет его состоянием. Подробности см. 3.3.

#### Описание

OUT и OUT NOT используются для управления состоянием заданного бита в соответствии с условиям исполнения.

OUT включает указанный бит в 1, когда условие исполнения = 1 и устанавливает его в 0, когда условие исполнения = 0. С битом TR команда работает не в конце командной линии, а в точке ветвления. Подробности см. 4.7.7.

OUT NOT включает указанный бит в 1, когда условие исполнения = 0 и устанавливает его в 0, когда условие исполнения = 1.

OUT и OUT NOT можно использовать для управления промежуточным результатом, устанавливая в 0 или 1 биты, которые заданы в качестве битов условия на релейно-контактной схеме, тем самым определяя условия срабатывания других команд. Это очень полезная функция, которая позволяет использовать сложный набор условий для управления одного рабочего бита и затем использовать данный рабочий бит для управления другими командами.

Промежутком времени, в котором бит находится в состоянии 0 или 1, можно управлять, комбинируя OUT и OUT NOT с TIM. Примеры см. 5.14.1.

#### Флаги

Данные команды не оказывают воздействия на флаги

### 5.9.2 DIFU и DIFD - Установка бита на 1 цикл (после фронта 0/1 и 1/0)

##### Обозначение на РКС



Номер инструкции		Операнды		
Контроллер	№	В		
		бит		
C200H□-CPU□□-E	13	IR, SR, HR, AR, LR		

Номер инструкции		Операнды		
Контроллер	Nr	B		
C200H-CPU-ZE	013	бит		
		IR, SR, HR, AR, LR		

**Обозначение на РКС**

— DIFD(Nr) B

Номер инструкции		Операнды		
Контроллер	Nr	B		
C200H-CPU-E	-	бит		
C200H-CPU-ZE	-	IR, SR, HR, AR, LR		
		IR, SR, HR, AR, LR		

**Ограничения**

Любой выходной бит можно использовать только с одной командой, которая управляет его состоянием. Подробности см. 3.3 Область IR.

**Описание**

DIFU и DIFD используются для установки указанного бита только на 1 цикл.

При выполнении DIFU сравнивается текущее условие исполнения с условием исполнения прошлого цикла. Если условие исполнения в прошлом цикле было = 0, а текущий = 1, DIFU устанавливает в 1 назначенный бит. Если условие исполнения в предыдущем цикле было = 1, то независимо от текущего состояния (0 или 1) DIFU либо установит заданный бит в 0, либо оставит его в 0 (т.е., если он уже был =0). Заданный бит будет в состоянии 1 только на 1 цикл, если только данная команда не выполняется каждый цикл (см. далее Предосторожности).

При выполнении DIFD сравнивается текущее условие исполнения с предыдущим условием исполнения. Если условие исполнения в прошлом цикле было = 1, а текущий = 0, DIFD устанавливает в 0 назначенный бит. Если условие исполнения в предыдущем цикле было = 0, а текущее либо 0, либо 1, DIFD либо установит заданный бит в 0, либо оставит его в 0 (т.е., если он уже был =0). Заданный бит будет в состоянии 1 только на 1 цикл, если только данная команда не выполняется каждый цикл (см. далее Предосторожности).

Данные команды используются в случае, если у команд версии фронта 0/1 (т.е. отмеченные @) нет, а желательно исполнение отдельной команды в течение одного цикла. Их можно также использовать с нормальной версией команд, у которых есть версия фронт 0/1, для упрощения программирования. Далее приведены примеры.

**Флаги**

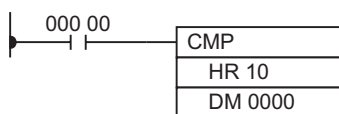
Данные команды не оказывают воздействия на флаги

**Предосторожности**

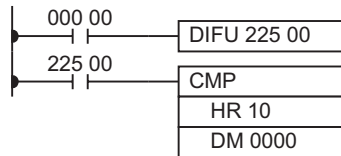
Операции DIFU и DIFD будут работать неустойчиво, когда команды запрограммированы между IL и ILC, между JMP и JME, или в подпрограммах. Подробности см. 5.10, 5.11, 5.23.

**Пример 1: Когда нет команды фронта 0/1**

На схеме А при выполнении CMP с условием исполнения = 1 постоянно будет сравниваться содержание двух слов операндов (HR 10 и DM 0000) и устанавливаться флаги арифметики (GR, EQ и LE) соответственно. Если условие исполнения остается 1, состояние флагов может быть изменено каждый цикл, если содержимое одного или двух операндов изменяется. Схема В - это пример, как можно использовать DIFU для того, чтобы обеспечить, чтобы CMP выполнялась только раз при включении требуемого условия исполнения в 1.



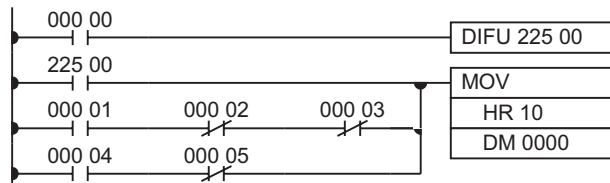
Адрес	Инструкция	Операнд
00000	LD	00000
00001	CMP	
		HR 10
		DM 0000



Адрес	Инструкция	Операнд
00000	LD	00000
00001	DIFU	22500
00002	LD	22500
00003	CMP	
		HR 10
		DM 0000

**Пример 2: Упрощение программирования**

Хотя у MOV есть форма фронта 0/1 (@), следующая схема была бы усложнена, если использовать эту форму, поскольку только одно условие, определяющее условие исполнения для MOV, требует формата фронта 0/1.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	DIFU	22500
00002	LD	22500
00003	LD	00001
00004	AND NOT	00002
00005	AND NOT	00003
00006	OR LD	-
00007	LD	00004
00008	AND NOT	00005
00009	OR LD	-
00010	MOV	
		HR 10
		DM 00

**5.9.3 SET и RESET (Установка и сброс)**

**Обозначение на ПКС**



Номер инструкции		Операнды	
Контроллер	Nr	B	бит
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR	

Номер инструкции		Операнды		
Контроллер	№г	В		
		бит		
C200H□-CPU□□-ZE	-	IR, SR, HR, AR, LR		

— RSET B

Номер инструкции		Операнды		
Контроллер	№г	В		
		бит		
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR		
C200H□-CPU□□-ZE	-	IR, SR, HR, AR, LR		

### Описание

SET включает битовый операнд в 1, когда условие исполнения = 1 и не влияет на состояние операнда, когда условие исполнения = 0. RSET устанавливает битовый операнд в 0, когда условие исполнения = 1 и не влияет на состояние операнда, когда условие исполнения = 0.

Операция SET отличается от OUT, поскольку команда OUT устанавливает битовый операнд в 0, когда условие исполнения = 0. Точно так же RESET отличается от OUT NOT тем, что команда OUT NOT устанавливает битовый операнд в 1, когда условие исполнения = 0.

### Предосторожности

Состояние битовых операндов команд SET и RESET, запрограммированных между IL и ILC или JMP и JME, не изменяется, когда выполняется условие модуля INTERLOCK или переходов (т.е. когда IL или JMP выполняется с условием исполнения 0).

### Флаги

Данные команды не оказывают воздействия на флаги

### Примеры

Следующие примеры демонстрируют разницу между OUT и SET/RESET. В первом примере (диаграмма А) IR 10000 установится в 1 или 0 когда IR 00000 установится в 1 или 0.

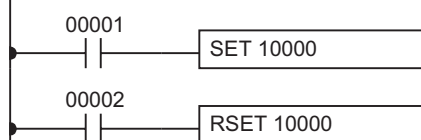
Во втором примере (диаграмма В) IR 10000 установится в 1, когда IR 00000 установится в 1 и останется =1 (даже если IR 00001 установится в 0) пока IR 00002 не станет = 1.

Диаграмма А



Адрес	Инструкция	Операнд
00000	LD	00000
00001	OUT	10000

Диаграмма В



Адрес	Инструкция	Операнд
00000	LD	00001
00001	SET	10000
00002	LD	00002
00003	RSET	10000

### 5.9.4 KEEP - Сохранить состояние

#### Обозначение на РКС



Номер инструкции		Операнды		
Контроллер	Nr	B		
		бит		
C200H□-CPU□□-E	-	IR, HR, AR, LR		
C200H□-CPU□□-ZE	-	IR, HR, AR, LR		

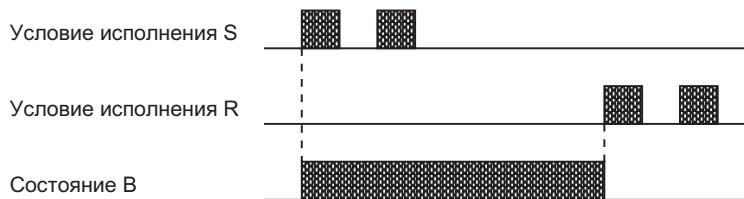
#### Ограничения

Любой выходной бит можно использовать только с одной командой, которая управляет его состоянием. Подробности см. 3.3 область IR.

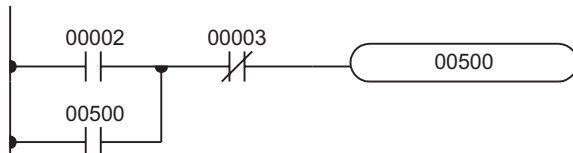
#### Описание

KEEP используется для поддержания состояния заданного бита исходя из двух условий. Данные условия обозначаются S и R. S - вход установки в 1, R - вход сброса. KEEP работает как реле с защелкой, которое устанавливается сигналом S и сбрасывается сигналом R.

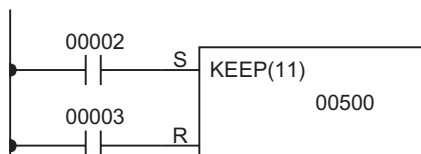
Когда S становится = 1, указанный бит устанавливается в 1 и остается до сброса, независимо от того, в каком состоянии находится S - 1 или 0. Когда R устанавливается в 1, указанный бит устанавливается в 0 и остается в состоянии 0 до установки, независимо от того, в каком состоянии находится R - 1 или 0. Взаимосвязь между результатами KEEP и состоянием бита KEEP показана на рисунке.



KEEP работает как биты самоподдержки, описанные в 4.8.3. Следующие две диаграммы будут работать аналогично, хотя та, которая использует KEEP, требует на одну команду меньше и сохраняет состояние даже в секции программы с INTERLOCK.



Адрес	Инструкция	Операнд
00000	LD	00002
00001	OR	00500
00002	AND NOT	00003
00003	OUT	00500



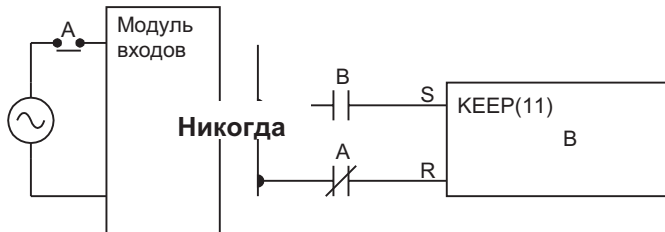
Адрес	Инструкция	Операнд
00000	LD	00002
00001	LD	00003
00002	KEEP	00500

**Флаги**

Данные команды не оказывают воздействия на флаги

**Предосторожности**

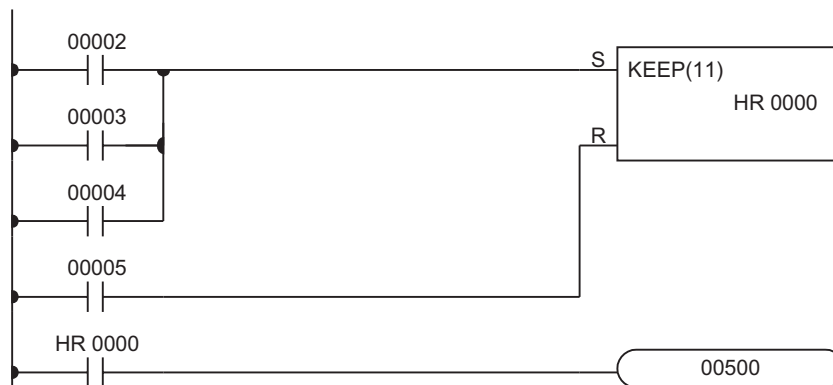
Будьте осторожны, когда линией сброса КЕЕР управляется нормально закрытое внешнее устройство. Не используйте входной бит с инверсным условием для линии сброса (R) КЕЕР, когда входное устройство работает на переменном токе. Задержка в снятии напряжения постоянного тока на ПК (по сравнению с переменным током устройства) может вызывать сброс указанного бита КЕЕР. Ситуация показана ниже.



Биты, используемые с КЕЕР, не сбрасываются в INTERLOCK. Подробности см. 5.10.

**Пример**

Если используется бит HR или AR, состояние бита сохранится даже при отключении питания. КЕЕР таким образом можно использовать для программирования битов, которые сохраняют состояние после перезапуска ПК после прерывания питания. Пример можно использовать для вывода предупреждающего сообщения после останова системы вследствие аварийной ситуации. Биты 00002, 00003 и 00004 будут установлены в 1 для индикации некоторых типов ошибок. Бит 00005 будет установлен в 1 для сброса предупреждающего сообщения. HR 0000, который устанавливается в 1, когда любой из этих трех битов указывает на аварийную ситуацию, служит для включения в 1 предупреждающий индикатор через 00500.



Адрес	Инструкция	Операнд
00000	LD	00002
00001	OR	00003
00002	OR	00004
00003	LD	00005
00004	KEEP	HR 0000
00005	LD	HR 0000
00006	OUT	00500

КЕЕР можно комбинировать с ТИМ для задержки включения битов в 1 и 0. Подробности см. 5.14.1.

## 5.9.5 TST и TSTN - Проверка бита

## Обозначение на РКС

TST
S
N
-

Номер инструкции		Операнды		
Контроллер	№	S	N	
		слово источник	номер бита	
C200H□-CPU□□-E				
C200H□-CPU□□-ZE	350	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM, #	-

TSTN
S
N
-

Номер инструкции		Операнды		
Контроллер	№	S	N	
		слово источник	номер бита	
C200H□-CPU□□-E				
C200H□-CPU□□-ZE	351	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM, #	-

## Ограничения

Любой бит может использоваться только в одной инструкции, контролирующей статус

## Описание

Выход TST находится в состоянии ON, когда контролируемый бит находится в состоянии ON, и переходит в состояние OFF, когда контролируемый бит находится в состоянии OFF.

Выход TSTN находится в состоянии ON, когда контролируемый бит находится в состоянии OFF, и переходит в состояние OFF, когда контролируемый бит находится в состоянии ON.

Номер бита N - двоично-десятичное число в диапазоне 0000..0015

## Предосторожности

TST и TSTN не могут использоваться как правосторонние инструкции, т.е. между выходом инструкции и правой панелью обязательно должна быть правосторонняя инструкция

## Флаги

**ER:** N не двоично-десятичное число в диапазоне 0000..0015.

Содержимое слова, являющегося косвенным адресом DM/EM не двоично-десятичное число или пересечена граница DM/EM.

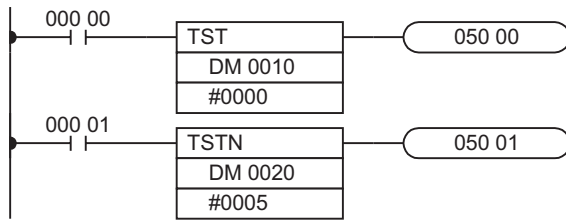
## Пример

Когда IR 00000 находится в состоянии ON, TST проверяет состояние контролируемого бита (бит 00 в DM 0010). Так как контролируемый бит находится в состоянии ON, IR 05000 также находится в состоянии ON.

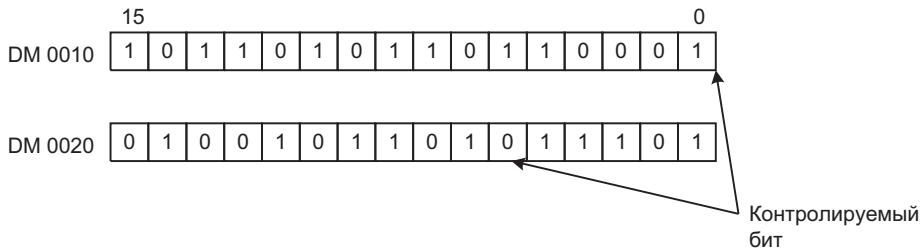
Когда IR 00001 находится в состоянии ON, TSTN проверяет состояние контролируемого бита (бит 05 в DM 0020). Так как контролируемый бит находится в состоянии OFF, IR 05000 находится в состоянии ON.



## 5.10 IL и ILC - Секция INTERLOCK и INTERLOCK CLEAR



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	TST	DM 0010
		#0000
00002	OUT	050 00
00003	LD	000 01
00004	TSTN	DM 0020
		#0005
00005	OUT	050 01



## 5.10 IL и ILC - Секция INTERLOCK и INTERLOCK CLEAR

### Обозначение на РКС

— IL

	Номер инструкции	Операнды		
Контроллер	№			
C200H-CPU□□-E	02			
C200H-CPU□□-ZE	002			

— ILC

	Номер инструкции	Операнды		
Контроллер	№			
C200H-CPU□□-E	03			
C200H-CPU□□-ZE	003			

### Описание

IL всегда используется совместно с ILC для создания секции INTERLOCK. Секции INTERLOCK служат для разрешения ветвления таким же образом, как это достигается битами TR, но обработка команд между IL и ILC отличается от обработки с битами TR, когда условие исполнения для IL = 0. Если условие исполнения для IL = 1, программа будет выполняться как написано, когда это условие, равное 1, служит для пуска всех командных линий после того места, где расположена команда IL до того места, где расположена ближайшая ILC. Базовые описания для обоих методов см. 4.3.8.

Если условие исполнения для IL = 0, секция INTERLOCK между IL и ILC будет обрабатываться, как указано в следующей таблице.

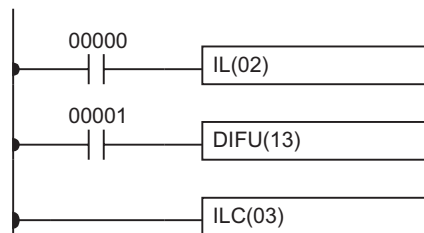
## 5.10 IL и ILC - Секция INTERLOCK и INTERLOCK CLEAR

Команда	Обработка
OUT и OUT NOT	Заданные биты устанавливаются в 0
SET и RSET	Сохраняется состояние битов
TIM TIMH	Сброс
TTIM	Сохраняется текущее значение
CNT, CNTR	Сохраняется текущее значение
KEEP	Сохраняется состояние бита
DIFU, DIFD	Не выполняются (смотри ниже)
Все другие команды	Команды не выполняются.

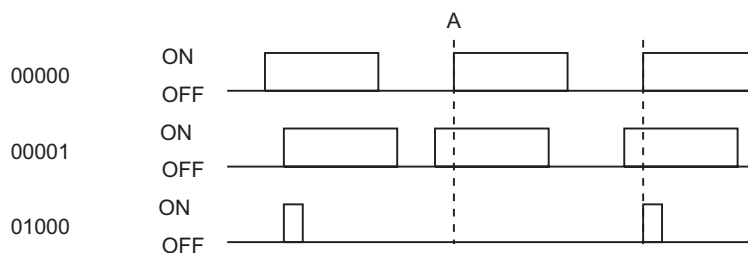
IL и ILC не обязательно использовать в паре. IL можно использовать несколько раз, каждая IL создает секцию INTERLOCK до ближайшей ILC. ILC можно использовать только когда для нее имеется хотя бы одна IL между ней и любой предыдущей ILC.

### DIFU, DIFU в секциях INTERLOCK

Изменение условий для DIFU или DIFD не запоминаются, если DIFU или DIFD находятся в секции INTERLOCK и условие исполнения для IL = 0. Когда DIFU или DIFD выполняются в секции INTERLOCK сразу после установки условия исполнения для IL в 1, условие для DIFU или DIFD будет сравниваться с тем условием исполнения, которое существовало до того, как секция INTERLOCK стала работать, (т.е. перед тем, как условие исполнения для IL стало = 0). ПКС и изменение состояния бита показаны на следующем рисунке. Секция INTERLOCK действует, пока 00000 = 0. Обратите внимание, что 01000 не включается в 1 в точке А, хотя 00001 стал = 0 и затем снова = 1.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	IL	
00002	LD	00001
00003	DIFU	01000
00004	ILC	



### Предосторожности

ILC должно быть расположено после одной или нескольких IL.

Хотя с одним ILC можно использовать столько IL, сколько необходимо, команды ILC нельзя использовать друг за другом, без хотя бы одной IL между ними, т.е. вложения невозможны.

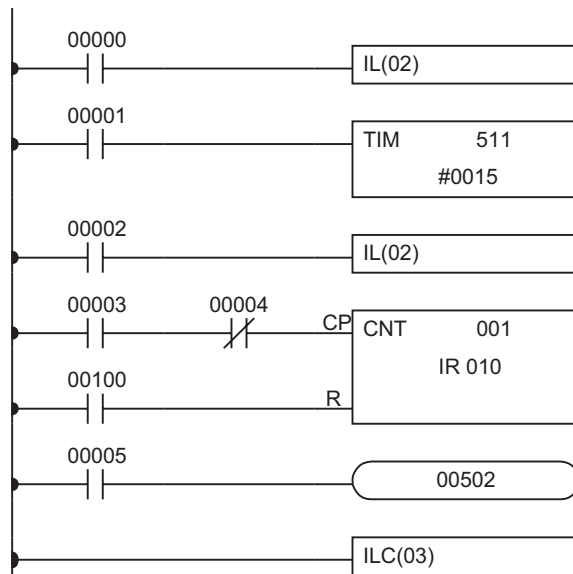
После выполнения ILC все группы между предыдущей ILC и текущей ILC убираются.

### Флаги

Данные команды не оказывают воздействия на флаги

**Пример**

В примере показано использование двух IL с одним ILC.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	IL	
00002	LD	00001
00003	TIM	511 # 0015
00004	LD	00002
00005	IL	
00006	LD	00003
00007	AND NOT	00004
00008	LD	00100
00009	CNT	001 010
00010	LD	00005
00011	OUT	00502
00012	ILC	

Когда условие исполнения для первой IL = 0, в TIM 127 будет занесено значение 1.5 с, CNT 001 не изменится, 00502 станет = 0. Когда условие исполнения для первой IL = 1, а условие исполнения для второй IL = 0, TIM 127 будет обрабатываться в зависимости от состояния 00002, CNT 001 не изменится, 00502 станет = 0. Когда условие исполнения для второй IL = 1, программа будет исполняться как написана.

**5.11 JMP и JME - Переход и конец перехода**

**Обозначение на РКС**

— JMP N

Контроллер	Номер инструкции	Операнды
	Nr	B бит
C200H□-CPU□□-E	04	I# (00..99)
C200H□-CPU□□-ZE	004	# (00..99)

— JME N

## 5.11 JMP и JME - Переход и конец перехода

Номер инструкции		Операнды		
Контроллер	№г	В		
		бит		
C200H-CPU-E	05	# (00..99)		
C200H-CPU-ZE	005	# (00..99)		

### Ограничения

Номера переходов 01 - 99 можно использовать только один раз в JMP и JME, т.е. каждый может использоваться для определения только одного перехода. Номер перехода 00 можно использовать сколько угодно раз.

### Описание

JMP всегда применяется совместно с JME для создания переходов, т.е. пропуска от одной до другой точки на релейно-контактной схеме. JMP определяет точку, с которой будет делаться переход. JME определяет адрес перехода. Когда условие исполнения для JMP = 1, перехода не происходит и программа выполняется без пропусков.. Когда условие исполнения для JMP = 0, происходит переход к JME с номером, таким же, как и у JMP, и далее выполняются команды, находящиеся после JME.

Если номер перехода для JMP находится в диапазоне 00 и 99, при выполнении переходов сразу же следует переход к команде JME с тем же номером, что и JMP без выполнения каких-либо команд между ними. Состояние таймеров, счетчиков и битов, используемых в OUT. биты, используемые в OUT NOT и все другие биты состояния, управляемые состояниями между JMP и JME не будут изменяться. Поскольку все команды между JMP и JME будут пропущены, номера переходов 01 - 99 используются для сокращения времени цикла.

Если номер перехода 00, ЦПУ будет искать ближайшую JME с номером 00. Для этого оно производит поиск по всей программе, вызывая удлинение времени цикла (когда условие исполнения = 0) по сравнению с другими переходами. Состояние таймеров, счетчиков, битов, используемых с OUT, битов, используемых с OUT NOT, и всех других битов, которые контролируются командами между JMP 00 и JME 00 не будет изменяться.. Переход JMP 00 всегда переходит к ближайшему следующему JME 00. Таким образом, можно использовать JMP 00 последовательно и завершать их одним JME 00. Однако нет смысла последовательно использовать JME 00, поскольку все переходы окончатся на первом JME 00.

### DIFU и DIFU в переходах

Хотя DIFU и DIFU предназначены для установки в 1 указанные биты на один цикл, они могут этого не сделать, если записаны между JMP и JME.

Как только либо DIFU, либо DIFU установит бит в 1, он останется в 1 до следующего выполнения DIFU или DIFU. При нормальном программировании, это значит - на следующем цикле. При переходе это значит следующий раз, когда нет пропуска от JMP к JME, т.е. если DIFU или DIFU установила бит в 1, а в последующих циклах происходит переход, так что DIFU или DIFU пропускаются, указанный бит останется в 1 до следующего раза, когда условие перехода станет = 1.

### Предосторожности

Когда JMP и JME используются не в паре, при контроле программы выдается сообщение об ошибке. Это сообщение появляется также, если JMP 00 или JME 00 используются не в паре, но программа будет выполняться без переходов.

### Флаги

Данные команды не оказывают воздействия на флаги

### Примеры

Примеры программ с переходами приведены в п. 4.7.8.

## 5.12 END - Конец программы

### Обозначение на ПКС

— END

		Номер инструкции	Операнды		
Контроллер		Nr			
C200H□-CPU□□-E		01			
C200H□-CPU□□-ZE		001			

### Описание

END требуется в качестве последней командной линии программы. Если есть подпрограммы, END помещается после последней подпрограммы Команды, записанные после END не выполняются. END можно поместить в любом месте программы, чтобы выполнять все команды до данного места., что иногда делается для отладки программы, но ее нужно удалить для выполнения остальной части программы.

Если в программе отсутствует END. никакие команды не выполняются и появляется сообщение NO END INST.

### Флаги

END сбрасывает в 0 флаги ER, CY, GR, EQ, LE.

## 5.13 NOP - Нет операции

		Номер инструкции	Операнды		
Контроллер		Nr			
C200H□-CPU□□-E		00			
C200H□-CPU□□-ZE		000			

### Описание

NOP при написании релейно-контактной схемы не требуется и не имеет символа для обозначения на релейно-контактной схеме. Когда NOP обнаруживается в программа, действий не производится и программа переходит к следующей команде. Когда память очищена перед программированием, во всех адресах записана команда NOP. Команда NOP вводится функциональным кодом 00.

### Флаги

NOP не оказывает воздействия на флаги

## 5.14 Команды таймеров и счетчиков

Команды TIM и TIMH - команды декрементирующего таймера, включающегося в 1, и требующие номеров ТС и заданного значения (SV).

CNT - команда декрементирующего счетчика, а команда CNTR - команда реверсивного счетчика. Обе требуют номер ТС и заданного значения (SV). Обе подключены к нескольким командным линиям, которые используются в качестве сигналов счетного входа (входов) и сброса.

Номер ТС нельзя задать более одного раза, т.е. если он уже задан в качестве определителя к команде таймера или счетчика, его нельзя больше использовать. После задания номера ТС можно использовать сколь угодно много раз в качестве операнда в других командах (не таймерах и не счетчиках).

Номера ТС задаются в диапазоне 000..511. При использовании номера таймера или счетчика в качестве определителя для команды таймера или счетчика префикс не требуется. Когда номер задан как таймер, его можно использовать с префиксом TIM для использования в качестве операнда для других команд. Префикс TIM применяется независимо от команды таймера, которой задавался таймер. Когда номер задан как счетчик, его можно использовать с префиксом CNT для использования в качестве операнда для других команд. Префикс CNT применяется также независимо от команды счетчика, которой задавался счетчик.

Номера ТС можно задать как операнды, требующие либо битовые, либо словные данные. Когда номер задан в качестве операнда, требующего битовых данных, данный номер получает доступ к биту, выполняющему функцию "флаг завершения", который указывает, что заданное время отработано/счет завершен, т.е. бит, нормально 0, который становится =1, когда заданное время отработано/счет завершен. Когда номер T/C задан в качестве операнда, требующего данных в виде слова, данный номер получает доступ к ячейке памяти, которая содержит текущее значение таймера или счетчика. Таким образом, текущее значение таймера или счетчика. Можно использовать в качестве операнда для команды SMP или других команд, для которых разрешена область ТС. Это достигается назначением номера ТС для доступа к области памяти, в которой хранится текущее значение.

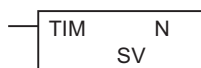
Обратите внимание, что TIM 000 используется и для задания команды TIM с номером 000, и для задания флага завершения для этого таймера, и для задания текущего значения этого таймера. Значение должно быть ясно из контекста, т.е. первой всегда идет команда, далее всегда битовый операнд и третьим всегда идет операнд - слово. То же самое верно для всех остальных номеров с префиксами TIM или CNT.

Заданное значение должна быть константой или адресом слова в области данных. Если в качестве слова источника задания задано слово области входов, указывающее на Модуль входов, подключения к этому Модулю должны быть сделаны таким образом, чтобы вводить задания с кодирующего колеса или другие аналогичные устройства.

Таймеры и счетчики, подключенные подобным образом, могут получить задание только из внешнего источника в режимах RUN или MONITOR. Все заданные значения, включая заданные извне, должны быть в двоично-десятичном виде.

### 5.14.1 TIM - Таймер

#### Обозначение на ПКС



Контроллер	Номер инструкции №	Операнды	
		N	SV
		номер таймера	заданное значение
C200H□-CPU□□-E	-	# (000..511)	IR, HR, AR, LR, DM, #
C200H□-CPU□□-ZE	-	# (000..511)	IR, SR, HR, AR, LR, DM, EM, #

#### Ограничения

Заданные значения лежат в диапазоне 000.0..999.9. Десятичная точка не вводится.

Каждый номер ТС можно использовать в качестве определителя только для одной команды таймера TIMER или счетчика COUNTER.

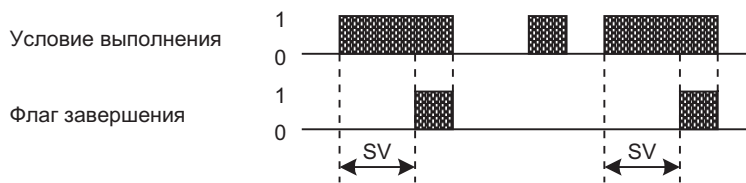
Номера ТС 000..015 нельзя задействовать командой TIM, если они нужны в команде TIMH. Подробности см. 5-15-2.

**Описание**

Таймер запускается, когда условие срабатывания устанавливается в 1 и сбрасывается (на заданное значение), когда условие срабатывания = 0. После запуска TIM отсчитывает время, вычитая по дискрете (в 0.1 мс) от задания.

Если условие срабатывания остается в 1 достаточно долго для отсчета текущего значения до нуля, флаг завершения данного номера устанавливается в 1 и остается в 1 до сброса таймера (т.е. когда условие станет = 0).

На диаграмме показано соотношение между условием отработки TIM и Флага завершения данного номера.



**Предосторожности**

Таймеры в секциях программы с INTERLOCK (сблокированные выходы) сбрасываются, когда условие исполнения IL = 0. Таймеры также сбрасываются при сбросе питания. Если нужен таймер, который не сбрасывался бы при таких условиях, биты импульсов времени в области SR можно подсчитать для задания таймеров, использующих CNT. Подробности см. 5.14.4.

Исполнение программы будет продолжено, даже если задание не двоично-десятичное число, но тогда время будет отсчитываться неверно.

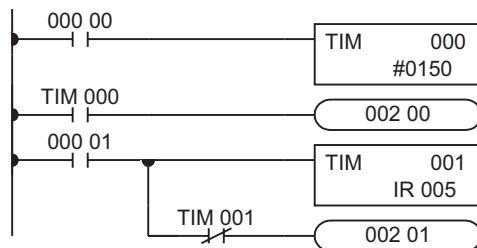
**Флаги**

**ER:** Заданные значения не в двоично-десятичном виде.

Косвенно адресуемые слова DM не существуют. (Содержимое слова \*DM не в двоично-десятичном виде, или превышена граница области DM)

**Пример 1: Основное применение**

В следующем примере показаны два таймера, у одного задание - константа, у другого - в слове входа 005. Бит 00200 включится в 1 после того, как 00000 включится в 1 и остается в 1 не менее 15 с. Когда 00000 сбрасывается в 0, таймер будет сброшен и 00200 будет сброшен в 0. Когда 00001 установится в 1, TIM 001 стартует с заданием, указанным в IR 005. Бит 00201 также включится в 1, когда 00001 включится в 1. Когда задание в 005 истечет, 00201 = 0. Этот бит также включится в 0, когда TIM 001 сброшен, независимо от того, истекло ли время задания.

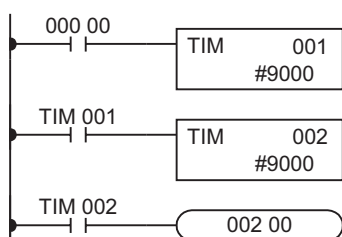


Адрес	Инструкция	Операнд
00000	LD	00000
00001	TIM	000
		# 0150
00002	LD	TIM 000

Адрес	Инструкция	Операнд
00003	OUT	00200
00004	LD	00001
00005	TIM	001
		005
00006	AND NOT	TIM 001
00007	OUT	00200

**Пример 2: Расширенные таймеры**

Есть 2 способа получения таймеров, работающих с величинами свыше 999.9 с. Один метод - программировать последовательные таймеры, когда флаг завершения одного служит для запуска следующего таймера. Простой пример с двумя таймерами на 900.0 с (15 минут), которые образуют таймер на 30 с.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	TIM	001
		# 9000
00002	LD	TIM 001
00003	TIM	002
		# 9000
00004	LD	TIM 002
00005	OUT	00200

В данном примере 00200 включатся в 1 через 30 мин. после включения 00000 в 1.

TIM можно также комбинировать с CNT, или CNT можно использовать для подсчета импульсов часов (области SR) для создания больших таймеров. Пример показан в 5.14.4.

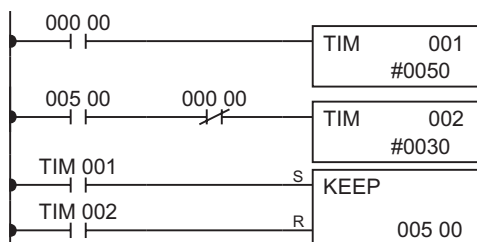
**Пример 3: Задержки 1/0**

TIM можно комбинировать с KEEP для задержки включения бита в 1 или 0 в зависимости от требуемого условия исполнения. KEEP описан в 5.9.4.

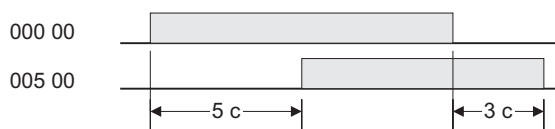
Для создания задержки используются флаги завершения двух таймеров для задания условий исполнения для установки и сброса бита, которым управляет KEEP. Бит, работу с которым нужно задержать, используется в KEEP. Включение в 1 и 0 бита для KEEP задерживается на величину задания двух TIM. Два задания при необходимости могут быть одинаковыми.

В следующем примере 00500 включится в 1 спустя 5.0 с после включения 00000 в 1 и затем сбросится в 0 спустя 3.0 с после того, как 00000 включится в 0. Необходимо использовать оба бита 00500 и 00000 для определения условий исполнения для TIM 002. 00000 необходим в инверсном условии для сброса TIM 002, когда 00000 включается в 1 и 00500 необходим для запуска TIM 002 (когда 00000 устанавливается в 0).



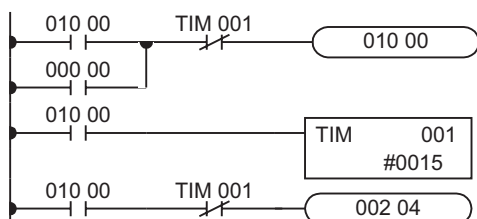


Адрес	Инструкция	Операнд
00000	LD	00000
00001	TIM	001
		# 0050
00002	LD	00500
00003	AND NOT	00000
00004	TIM	002
		# 0030
00005	LD	TIM 001
00006	LD	TIM 002
00007	KEEP	00500

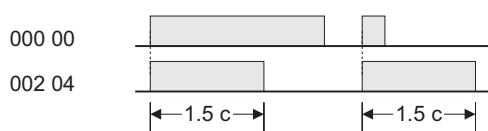


#### Пример 4: Биты для одного включения

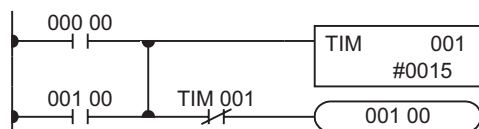
Промежуток времени, в течение которого бит находится в состоянии 1 или 0, можно регулировать, комбинируя TIM с OUT или OUT NOT. В следующей диаграмме демонстрируется такая возможность. В примере 00204 останется = 1 на 1.5 с после включения 00000 в 1 независимо от времени, когда 00000 остается в 1. Это достигается использованием 01000 в качестве бита самоподдержки, выполняющегося по 00000 и включающий 00204. Когда TIM 001 = 1 (т. е. после истечения задания TIM 001), 00204 = 0 с помощью TIM 001 (т. е. TIM 001 будет = 1, что создаст условие исполнения 0 для OUT 00204).



Адрес	Инструкция	Операнд
00000	LD	01000
00001	AND NOT	TIM 001
00002	OR	00000
00003	OUT	01000
00004	LD	01000
00005	TIM	001
		# 0015
00006	LD	01000
00007	AND NOT	TIM 001
00008	OUT	00204



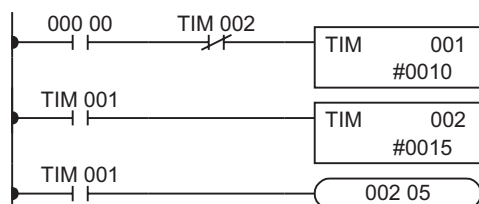
Следующий таймер на 1 импульс можно использовать для сохранения памяти.



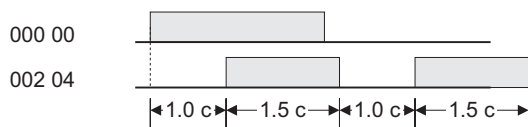
Адрес	Инструкция	Операнд
00000	LD	00000
00001	OR	00100
00002	TIM	001
		# 0015
00003	AND NOT	TIM 001
00004	OUT	00100

#### Пример 5: “Мигающие” биты

Можно запрограммировать включение битов в 1 и 0 через равные интервалы времени после срабатывания условия исполнения (путем использования команды TIM дважды). Один TIM служит для включения в 1 и 0 указанный бит, т. е. флаг завершения этого таймера включает указанный бит в 1 и 0. Другой таймер служит для управления операциями первого TIM, т. е. когда флаг завершения первого TIM = 1, второй TIM запускается и когда флаг завершения второго TIM = 1, стартует первый TIM.



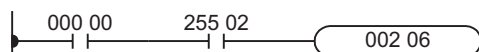
Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND	TIM 002
00002	TIM	001
		# 0010
00003	LD	TIM 001
00004	TIM	002
		# 0015
00005	LD	TIM 001
00006	OUT	00205



Более простой, но менее гибкий метод создания битов импульсов - это логическое сложение одного из импульсов часов области SR с условием исполнения, которое должно быть в 1 при работе импульсов. Хотя это метод не использует TIM, он включен в данную главу для сравнения. Этот метод ограничен, поскольку время 1 и 0 должно быть одинаковым, и зависит от импульсов области SR.

В следующем примере используется импульс 1-с (25502), так что 00206 будет = 1 и 0 каждую секунду, т. е. он будет = 1 в течение 0.5 с и = 0 в течение 0.5 с. точный отсчет

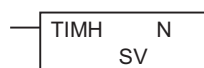
времени и исходное состояние 00206 будет зависеть от состояния импульсов при включении 00000 в 1.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND	25502
00002	OUT	00206

### 5.14.2 ТИМН - Высокоскоростной таймер

#### Обозначение на РКС



Контроллер	Номер инструкции	Операнды	
		N	SV
	№	номер таймера	заданное значение (слово, BCD)
C200H-CPU-E	15	# (000..511)	IR, HR, AR, LR, DM, #
C200H-CPU-ZE	015	# (000..511)	IR, SR, HR, AR, LR, DM, EM, #

#### Ограничения

Заданные значения лежат в диапазоне 00.00..99.99. (Хотя можно ввести 00.00 и 00.01, 00.00 отключит таймер, т.е. сразу включит флаг ИСПОЛНЕНО, а 00.01 не будет надежно считан.) Десятичная точка не вводится.

Каждый номер ТС можно использовать в качестве определителя только для одной команды таймера TIMER или счетчика COUNTER.

Если время цикла превышает 10 мс., используйте ТС 000..ТС 015. Текущие значения и флаги завершения таймеров 000..015 обновляются каждые 10 мс, но текущие значения таймеров 016..511 обновляются каждый раз, когда в программе выполняется ТИМН.

#### Описание

ТИМН работает так же, как и ТИМ, за исключением того, что дискрета ТИМН..0.01 с.

Время цикла влияет на точность ТИМН при использовании ТС 016..ТС 511. Если время цикла превышает 10 мс., используйте ТС 000..ТС 015.

Подробности о работе и примеры ТИМ см. 5-14-1. За исключением вышеизложенного, команды работают аналогично.

#### Предосторожности

Таймеры в секциях программы с INTERLOCK (сблокированные выходы) сбрасываются, когда условия для IL = 0. Таймеры также сбрасываются при сбросе питания. Если нужен таймер, который не сбрасывался бы при таких условиях, биты импульсов времени в области SR можно подсчитать для задания таймеров, использующих CNT. Подробности см. 5-14.4.

Исполнение программы продолжится, даже если используется не двоично-десятичное значение задание, но время будет отсчитываться неверно.

#### Флаги

**ER:** Заданные значения не в двоично-десятичном виде.

Косвенно адресуемые слова DM не существуют. (Содержимое слова \*DM не в двоично-десятичном виде, или превышена граница области DM)

## 5.14.3 ТТИМ - Суммирующий таймер

## Обозначение на РКС

ТТИМ
N
SV
RB

Контроллер	Номер инструкции №	Операнды		
		N номер таймера	SV заданное значение (слово, BCD)	RB бит сброса
C200H-CPU-E	87	# (000..511)	IR, HR, AR, LR, DM	IR, SR, HR, AR, LR
C200H-CPU-ZE	087	# (000..511)	IR, SR, HR, AR, LR, DM, EM, #	IR, SR, HR, AR, LR

## Ограничения

SV должен быть двоично-десятичным числом 0000..9999 (000.0.... 999.9 с). Десятичная точка не вводится.

Каждый номер ТС можно использовать в качестве определителя только в одной команде TIMER или COUNTER.

## Описание

ТТИМ служит для создания таймера, который инкрементирует текущее значение каждые 0.1 с (до диапазона 0.1..999.9 с). ТТИМ увеличивает текущее значение от 0 с дискретой 0.1 с. Точность ТТИМ - +0.0/-0.1 с. Таймер ТТИМ будет производить отсчет, пока условие исполнения = 1, пока не достигнет значения задания или не включится в 1 бит RB для сброса таймера. Таймеры ТТИМ будут производить отсчет в каждом цикле только тогда, когда к ним есть обращение, т. е. они не считают, но сохраняют текущие значения в секциях INTERLOCK или JUMP.

Текущие значения и флаги завершения таймеров 000..015 обновляются каждые 10 мс, а текущие значения таймеров 016..511 обновляются каждый раз, когда в программе встречаются ТТИМ.

## Предосторожности

Текущее значение суммирующего таймера в секциях INTERLOCK сохраняются, пока условие исполнения для IL = 0. В отличие от таймеров и высокоскоростных таймеров, суммирующие в секциях JUMP не продолжают отсчет времени, а сохраняют свое текущее значение.

Прерывание питания сбрасывает таймеры.

Суммирующие таймеры не перезапускаются после отсчета времени, если не изменяется текущее значение на значение, которое меньше задания, или таймер не сброшен входом сброса.

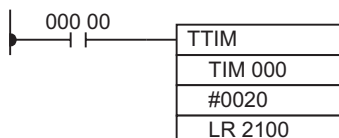
Иногда требуется задержка на цикл для включения в 1 флага завершения после отработки времени.

## Флаги

**ER:** Содержимое слова \*DM не в двоично-десятичном виде.  
Задание не в двоично-десятичном виде.

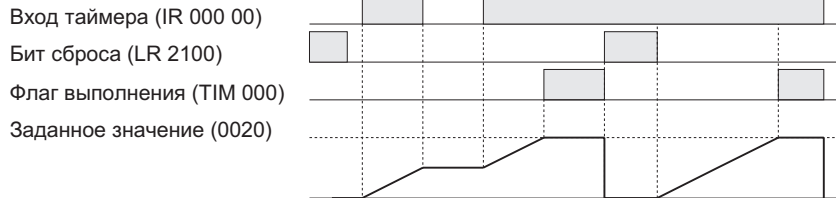
## Пример

Следующая схема иллюстрирует отношение между всеми условиями исполнения для суммирующего таймера с заданием 0.2 с, его текущим значением и флагом завершения.



Адрес	Инструкция	Операнд
00000	LD	00000

Адрес	Инструкция	Операнд
00001	TTIM	
		TIM 000
		# 0020
		LR 2100



### 5.14.4 CNT - Счетчик

#### Обозначение на ПКС

CP	CNT	N
R		SV

Контроллер	Номер инструкции	Операнды	
		N	SV
	№	номер счетчика	заданное значение
C200H□-CPU□□-E	-	# (000..511)	IR, HR, AR, LR, DM, #
C200H□-CPU□□-ZE	-	# (000..511)	IR, SR, HR, AR, LR, DM, EM, #

#### Ограничения

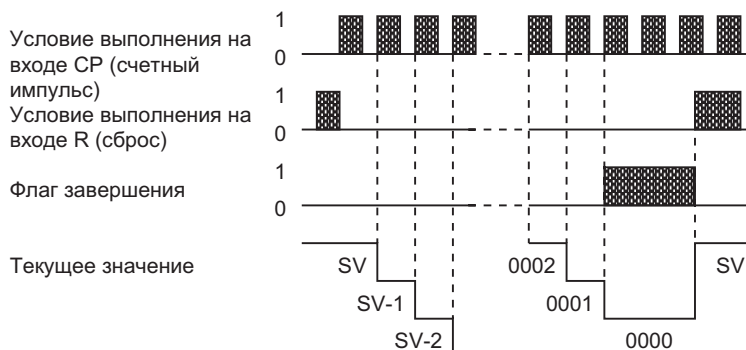
Каждый номер ТС можно использовать в качестве определителя только для одной команды таймера TIMER или счетчика COUNTER.

#### Описание

CNT служит для отсчета вниз от заданного значения, когда условие исполнения на счетном входе, CP, изменяется из 0 в 1, т.е. текущее значение будет декрементировано (уменьшено на 1) при текущем состоянии счетного выхода = 1 и состоянием в прошлом цикле = 0. Если условие исполнения не изменяется или изменяется из 1 в 0, текущее значение счетчика не изменяется. Флаг завершения устанавливается в 1, когда текущее значение достигает 1 и остается 1 до сброса счетчика.

Счетчик сбрасывается входом сброса, R.. Когда R изменяется из 0 на 1, текущее значение сбрасывается до задания. Пока R = 1, текущее значение не декрементируется. Отсчет вниз от задания начнется снова, когда R станет = 0. Текущее значение не будет сброшено в секциях заблокированных входов или при прерывании питания.

На диаграмме показаны изменения условия отработки, Флаг завершения и текущее значение счетчика. Разная высота линии текущего значения служит только для того, чтобы показать изменение текущего значения.



**Предосторожности**

Выполнение программы продолжится, если даже задание введено не в двоично-десятичном виде, но тогда задание будет неверным.

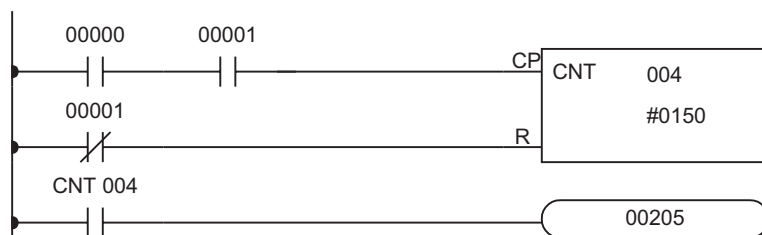
**Флаги**

**ER:** Задание не в двоично-десятичном виде.

Косвенно адресуемые слова DM не существуют. (Содержимое слова \*DM не в двоично-десятичном виде, или превышена граница области DM)

**Пример 1: Базовое применение**

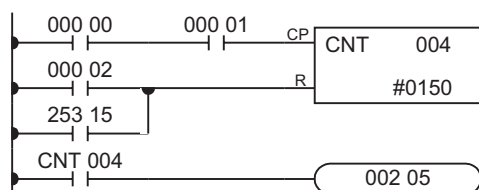
В следующем примере текущее значение будет декрементироваться, когда оба 00000 и 00001 установлены в 1, и 00002 = 0 и либо 00000, либо 00001 были равны 0 при выполнении CNT 004 последний раз. Когда будет отсчитано 150 импульсов (т. е. когда текущее значение станет = 0), 00205 установится в 1.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND	00001
00002	LD	00002
00003	CNT	004
		# 0150
00004	LD	CNT 004
00005	OUT	00205

Здесь, 00000 можно использовать для управления (рабочий режим CNT), а 00001 можно использовать как бит, изменение состояния которого 0 или 1 подсчитываются.

Ранее показанный CNT можно модифицировать для перезапуска каждый раз при включении питания на ПК. Это делается с помощью флага первого цикла SR 25315 для сброса CNT, как показано далее.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND	00001
00002	LD	00002
00003	OR	25315
00004	CNT	004
		# 0150
00005	LD	CNT 004
00006	OUT	00205

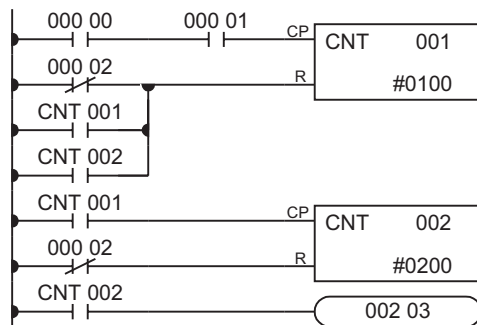
**Пример 2: Расширенный счетчик**

Счетчики, способные считать более 9 999, можно запрограммировать с помощью одного CNT для подсчета числа раз, когда второй CNT досчитал от задания до 0.

В следующем примере 0000 служит для разрешения работы счетчика CNT 001. CNT 001, когда 00000 = 1, считает вниз количество переключений из 0 в 1 входа 00001. CNT 001 сбрасывается флагом завершения, т. е. снова начинает отсчет, как только текущее значение достигнет 0. CNT 002 считает число включений в 1 флага завершения CNT 001. Бит 00002 служит в качестве сброса для всего расширенного счетчика, сбрасывая оба CNT 001 и CNT 002, когда он = 0. Флаг завершения CNT 002 также служит для сброса CNT 001 для запрещения операции CNT 001, если задание CNT 002 достигнуто, до тех пор, пока расширенный счетчик не сбросится битом 00002.

Поскольку в этом примере задание для CNT 001 и для CNT 002 = 200, флаг завершения для CNT 002 включится в 1, когда 100 x 200 или 20 000 переходов 0\1 будет отсчитано в 00001. Это приведет к включению 00203 в 1.

CNT можно использовать последовательно столько раз, сколько нужно для создания счетчиков, достаточных для отсчета любой требуемой величины.



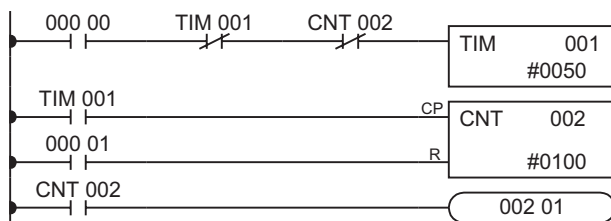
Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND	00001
00002	LD NOT	00002
00003	OR	CNT 001
00004	OR	CNT 002
00005	CNT	001 # 0100
00006	LD	CNT 001
00007	LD NOT	00002
00008	CNT	002 # 0200
00009	LD	CNT 002
00010	OUT	00203

### Пример 3: Расширенные таймеры

CNT можно использовать для создания расширенных таймеров двумя способами: комбинируя TIM с CNT и подсчетом импульсов битов области SR.

В следующем примере CNT 002 отсчитывает, сколько раз TIM 001 достигает нуля от задания. Флаг завершения TIM 001 служит для сброса TIM 001, чтобы это продолжалось, и CNT 002 отсчитывает то число раз, сколько флаг завершения TIM 001 устанавливается. TIM 001 также сбрасывается флагом завершения CNT 002, так что расширенный таймер не запустится снова до сброса CNT 002 битом 00001, что служит сбросом для всего расширенного таймера.

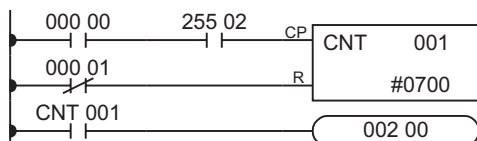
Так как в данном примере задание для TIM 001 = 5.0 секунд и задание для CNT 002 = 100, флаг завершения CNT 002 будет 1 после 5 секунд x 100 раз, т. е. 500 секунд (8 мин. 20 сек) истекло. Это приведет к включению в 1 00201.



Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND NOT	TIM 001
00002	AND NOT	CNT 002
00003	TIM	001
		# 0050
00004	LD	TIM 001
00005	LD	00001
00006	CNT	002
		# 0100
00007	LD	CNT 002
00008	OUT	00201

В следующем примере CNT 001 отсчитывает, сколько раз импульс 1 сек (25502) переключился из 0 в 1. Здесь 00000 служит для управления.

Поскольку в данном примере задание для CNT 001 = 700, флаг завершения CNT 002 = 1 по истечении 1 сек x 700 раз, или 11 мин. 40 сек. Это приведет к включению 00202 в 1.

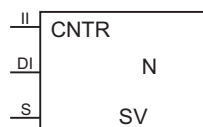


Адрес	Инструкция	Операнд
00000	LD	00000
00001	AND	25502
00002	LD NOT	00001
00003	CNT	001
		# 0700
00004	LD	CNT 001
00005	OUT	00202

**Замечание** Самые короткие импульсы необязательно создадут точные таймеры, поскольку их короткие состояния 1 могут быть прочитаны неточно при длинных циклах. Особенно не подходят импульсы длительностью 0.02 и 0.1 с.

### 5.14.5 CNTR - Реверсивный счетчик

Обозначение на РКС



Номер инструкции		Операнды	
Контроллер	№	N	SV
		номер счетчика	заданное значение (слово, BCD)
C200H□-CPU□□-E	12	# (000..511)	IR, HR, AR, LR, DM, #



Номер инструкции		Операнды		
Контроллер	№г	N	SV	
		номер счетчика	заданное значение (слово, BCD)	
C200H□-CPU□□-ZE	012	# (000..511)	IR, SR, HR, AR, LR, DM, EM, #	

### Ограничения

Каждый номер ТС можно использовать в качестве определителя только для одной команды таймера TIMER или счетчика COUNTER.

### Описание

CNTR - реверсивный, двухсторонний кольцевой счетчик, т.е. он служит для счета от 0 до задания в зависимости от изменений двух условий исполнения, на входе инкрементирования (II) и входе декрементирования (DI).

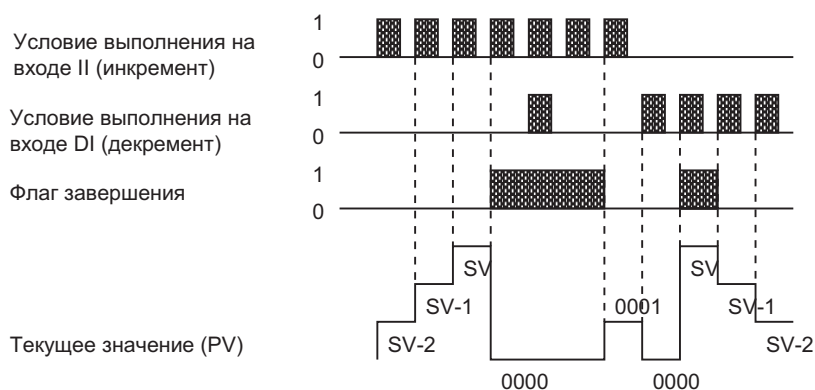
Текущее значение инкрементируется при выполнении CNTR с текущим условием исполнения входа II = 1, а прошлое значение II было = 0. Текущее значение декрементируется при выполнении CNTR с текущим условием исполнения DI = 1, а прошлое значение II было = 0. Если условия исполнения на обоих входах изменились из 0 в 1, текущее значение счетчика не изменяется.

Если условия на II и DI не изменяются или изменяется из 1 в 0, текущее значение счетчика не изменяется.

При декрементировании от 0000 текущее значение установится в заданное, а флаг ИСПОЛНЕНО устанавливается в 1 до следующего декрементирования счетчика. При инкрементировании от задания текущее значение установится в 0, а флаг ИСПОЛНЕНО устанавливается в 1 до следующего инкрементирования счетчика.

Счетчик CNTR сбрасывается входом сброса, R.. Когда R изменяется из 0 на 1, текущее значение сбрасывается в 0. Пока R = 1, текущее значение не декрементируется и не инкрементируется. Счет начнется снова, когда R станет = 0. Текущее значение CNTR не будет сброшен в секциях заблокированных выходов INTERLOCK или при прерывании питания.

На диаграмме показаны изменения условий исполнения на входах II и DI, Флаг ИСПОЛНЕНО и текущее значение счетчика при операциях счетчика (сброса, счет начинается с нуля). Разная высота линии текущего значения служит только для того, чтобы показать изменение текущего значения.



### Предосторожности

Выполнение программы продолжится, если даже задание введено не в двоично-десятичном виде, но тогда задание будет неверным.

### Флаги

**ER:** Задание не в двоично-десятичном виде.

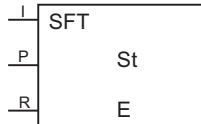
Косвенно адресуемые слова DM не существуют. (Содержимое слова \*DM не в двоично-десятичном виде, или превышена граница области DM)

## 5.15 Команды сдвига

Все команды, описанные в данной главе, используются для сдвига данных, но на разные значения и в разных направлениях. Первая команда сдвига, SFT, вдвигает условие исполнения в регистр сдвига; Остальные команды сдвигают данные, которые уже находятся в памяти.

### 5.15.1 SFT - Сдвиговый регистр

#### Обозначение на ПКС



Контроллер	Номер инструкции Nr	Операнды	
		St первое слово	E последнее слово
C200H□-CPU□□-E	10	IR, SR, HR, AR, LR	IR, SR, HR, AR, LR
C200H□-CPU□□-ZE	010	IR, SR, HR, AR, LR	IR, SR, HR, AR, LR

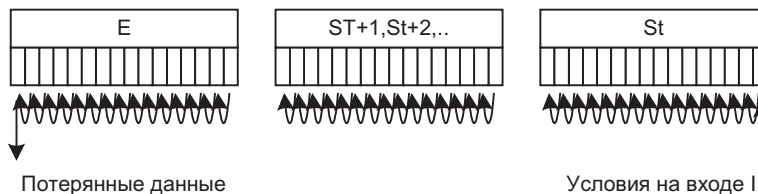
#### Ограничения

E должно быть больше либо равно St, а St и E должны быть в одной области данных.

Если битовый адрес в одном из слов, используемых при сдвиге регистра, также задействован в команде, которая управляет состоянием индивидуального бита (напр. OUT, KEEP) появится сообщение об ошибке COIL DUPL при контроле программы на синтаксис с программатора или других устройств программирования. Программа, однако, будет выполняться. Примеры программирования см. пример 2.

#### Описание

SFT управляется тремя условиями, I, P, и R. Если SFT выполняется и условие исполнения  $P = 1$  и было  $= 0$  при прошлом заходе  $R = 0$ , тогда условие I переместится в самый правый (младший) бит регистра сдвига, заданный между St и E, т.е. если  $I = 1$ , в регистр заносится 1, если  $I = 0$ , в регистр заносится 0. Когда I заносится в регистр, все биты, бывшие в регистре, сдвигаются влево, а самый старший бит регистра теряется.



Условие на P действует как команда фронта 0/1, т.е. I. будет внесено в регистр, только когда P изменялось с 0 на 1. Если условие P не изменялось или изменилось с 1 на 0, регистр сдвига не изменяется.

St определяет младшее слово регистра сдвига, E - старшее. Регистр сдвига включает эти слова и слова между ними. Одно и то же слово можно задать и как старшее, и как младшее, т.е. создать регистр сдвига 16 бит (1 слово).

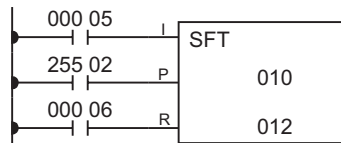
Когда условие  $R = 1$ , все биты в регистре устанавливаются в 0, и регистр сдвига не будет работать, пока R не станет  $= 0$ .

#### Флаги

Команда SFT не воздействует на флаги.

#### Пример 1: Основное применение

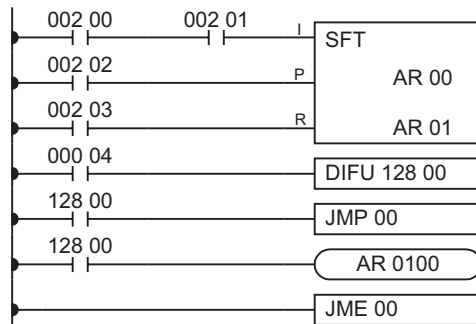
В следующем примере используется бит импульса 1 с (25502), чтобы условие, определяемое 00005, вдвигалось в регистр сдвига из трех слов IR 010..IR 012 каждую секунду.



Адрес	Инструкция	Операнд
00000	LD	00005
00001	LD	25502
00002	LD	00006
00003	SFT	
		010
		012

### Пример 2: Биты управления в регистре сдвига

Следующая программа служит для управления состоянием 17-го бита в регистре сдвига AR 00..AR 01. Когда нужно установить бит 17, включается 00004. В этом случае переход JMP 00 не выполняется на один этот цикл, и AR 0100 (бит 17) будет включен в 1. Когда  $12800 = 0$  (т. е. на все время, исключая первый цикл после изменения 00004 с 0 на 1), переход выполняется, и состояние AR 0100 не изменится.



Адрес	Инструкция	Операнд
00000	LD	00200
00001	AND	00201
00002	LD	00202
00003	LD	00203
00004	SFT	
		AR 00
		AR 01
00005	LD	00004
00006	DIFU	12800
00007	LD	12800
00008	JMP	00
00009	LD	12800
00010	OUT	AR 0100
00011	JME	00

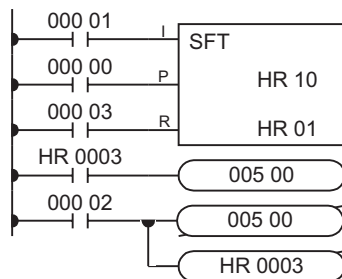
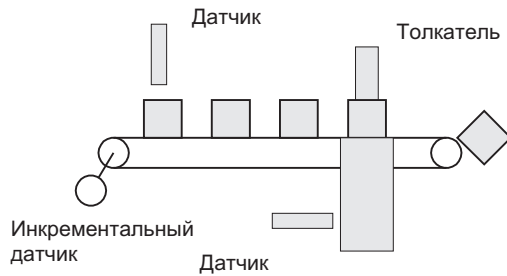
Когда бит - часть сдвигового регистра - используется в команде OUT (или любой другой команде, которая управляет состоянием бита), во время проверки программы появляется признак ошибки, но программа будет выполняться корректно (т. е. как написано).

### Пример 3: Управляющее действие

Следующая программа управляет конвейерной линией, показанной ниже, в которой обнаруживаются бракованные изделия и сталкиваются вниз по лотку. Для этого условие исполнения, определяемое входом с первого датчика (00001) сохраняется в регистре

сдвига: 1 для кондиционных изделий, 0 для бракованных. Скорость конвейера настраивается так, что HR 0003 сдвигового регистра служит для запуска толкателя (00500) когда бракованное изделие подходит к толкателю, т. е. когда HR 0003 = 1, 00500 = 1 для пуска толкателя.

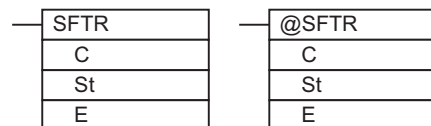
Программа настраивается так, что вращающийся датчик (0000) управляет исполнением SFT командой DIFU, вращающийся датчик настраивается для на включение в 1 и 0 каждый раз, когда изделие проходит первый датчик. Другой датчик (00002) служит для обнаружения бракованных изделий в лотке, чтобы выход толкателя и HR 0003 регистра сдвига можно было сбросить в случае надобности.



Адрес	Инструкция	Операнд
00000	LD	00001
00001	LD	00000
00002	LD	00003
00003	SFT	HR 00
		HR 01
00004	LD	HR 0003
00005	OUT	00500
00006	LD	00002
00007	OUT NOT	00500
00008	OUT NOT	HR 0003

### 5.15.2 SFTR - Реверсивный регистр сдвига

Обозначение на ПКС



Контроллер	Номер инструкции Nr	Операнды		
		C	St	E
		слово состояния	первое слово	последнее слово
C200H-CPU-E	84	IR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM
C200H-CPU-ZE	084	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, DM, EM

#### Ограничения

St и E должны лежать в одной области данных. ST должно быть меньше либо равно E.

**Описание**

SFTR служит для создания регистра сдвига из одного или нескольких слов, который может сдвигать данные и вправо и влево. Для создания регистра из одного слова задайте одинаковыми ST и E. В слове управления направление сдвига, состояние, которое должно быть задано регистру, импульс сдвига и ввод сброса. Слово управления имеет следующий вид:



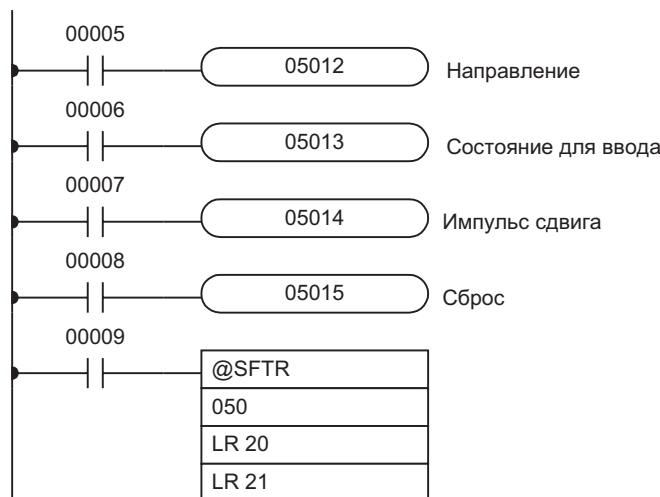
Данные в регистре сдвига будут сдвигаться на один бит в направлении, указанном битом 12, выталкивая один бит в CY и принимая с другой стороны состояние бита 13, когда, при условии исполнения = 1, SFTR выполняется (до тех пор, пока бит сброса = 0 и пока бит 14 = 1). Если SFTR выполняется с условием 0 или с битом 14 = 0, регистр сдвига останется в прежнем состоянии. Если SFTR выполняется с условием 1 и бит сброса (бит 15) = 0, все слова регистра и CY сбросятся в 0.

**Флаги**

- ER:** Слова St и E лежат в разных областях или St больше E. Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или превышена граница области DM)
- CY:** Получает состояние бита 00 слова St или бита 15 слова E в зависимости от направления сдвига.

**Пример**

В примере IR 00005, IR 00006, IR 00007 и IR 00008 используются для управления битами слова С, используемого в @SFTR. Регистр сдвига - LR 20..LR 21, и он управляется IR 00009.

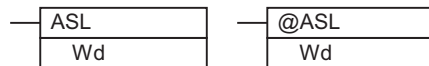


Адрес	Инструкция	Операнд
00000	LD	00001
00001	LD	00000
00002	LD	00003
00003	SFT	
00004	LD	HR 0003

Адрес	Инструкция	Операнд
00005	OUT	00500
00006	LD	00002
00007	OUT NOT	00500
00008	OUT NOT	HR 0003
00009	@SFTR	
		050
		LR 20
		LR 21

### 5.15.3 ASL - Арифметический сдвиг влево

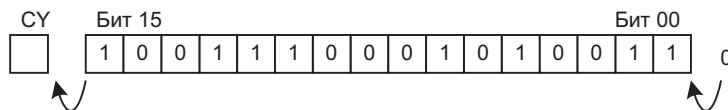
#### Обозначение на ПКС



Номер инструкции		Операнды		
Контроллер	№	Wd		
		слово сдвига		
C200H-CPU□□-E	25	IR, SR, HR, AR, LR, DM		
C200H-CPU□□-ZE	025	IR, SR, HR, AR, LR, DM, EM		

#### Описание

Когда условие исполнения = 0, ASL не выполняется. Когда условие исполнения = 1, ASL помещает 0 в бит 00 Wd, сдвигает на 1 биты слова Wd влево, и переносит состояние бита 15 в CY.

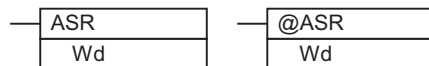


#### Флаги

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или превышена граница области DM)
- CY:** Получает состояние бита 15
- EQ:** 1, когда содержимое Wd равен нулю, в противном случае = 0.
- N:** 1, когда 1 вдвигается в бит 15 WD.

### 5.15.4 ASR - Арифметический сдвиг вправо

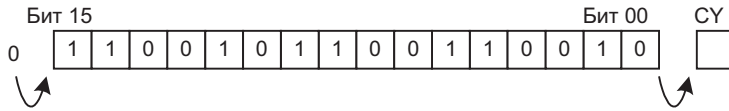
#### Обозначение на ПКС



Номер инструкции		Операнды		
Контроллер	№	Wd		
		слово сдвига		
C200H-CPU□□-E	26	IR, SR, HR, AR, LR, DM		
C200H-CPU□□-ZE	026	IR, SR, HR, AR, LR, DM, EM		

#### Описание

Когда условие исполнения = 0, ASR не выполняется. Когда условие исполнения = 1, ASR помещает 0 в бит 15 Wd, сдвигает на 1 биты слова Wd вправо, и переносит состояние бита 00 в CY.



**Предосторожности**

В бит 00 будет заноситься нуль каждый цикл, если используется нормальная версия ASR (не фронта 0/1). Используйте версию фронта 0/1 (@ASR) или комбинируйте ASR с DIFU. DIFU для сдвига только один раз.

**Флаги**

**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или превышена граница области DM)

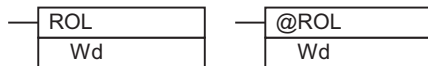
**CY:**

Получает состояние бита 00

**EQ:** 1, когда содержимое Wd равен нулю, в противном случае = 0.

**5.15.5 ROL - Циклический сдвиг влево**

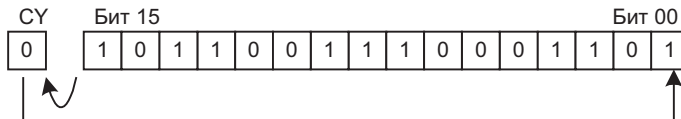
**Обозначение на ПКС**



Номер инструкции		Операнды		
Контроллер	№	Wd		
C200H-CPU-E	27	слово сдвига		
C200H-CPU-ZE	027	IR, SR, HR, AR, LR, DM		
		IR, SR, HR, AR, LR, DM, EM		

**Описание**

Когда условие исполнения = 0, ROL не выполняется. Когда условие исполнения = 1, ROL сдвигает на 1 все биты слова Wd влево, помещает CY в бит 00 Wd, и переносит состояние бита 15 в CY.



**Предосторожности**

Для установки состояния CY в 1 используйте команду STC или CLC для очистки CY перед командой сдвига ROL. чтобы быть уверенным, что в CY содержатся правильное значение.

Состояние CY очищается в конце каждого цикла (когда выполняется END).

**Флаги**

**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или превышена граница области DM)

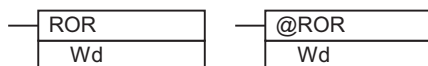
**CY:** Получает состояние бита 15

**EQ:** 1, когда содержимое Wd равен нулю, в противном случае = 0.

**N:** 1, когда 1 вдвигается в бит 15 WD.

**5.15.6 ROR - Циклический сдвиг вправо**

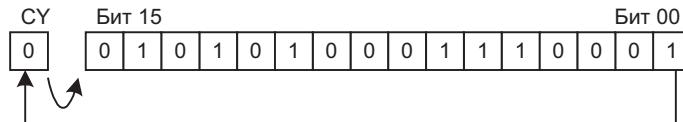
**Обозначение на ПКС**



Номер инструкции		Операнды		
Контроллер	№	Wd		
		слово сдвига		
C200H□-CPU□□-E	28	IR, SR, HR, AR, LR, DM		
C200H□-CPU□□-ZE	028	IR, SR, HR, AR, LR, DM, EM		

**Описание**

Когда условие исполнения = 0, ROR не выполняется. Когда условие исполнения = 1, ROR сдвигает на 1 все биты слова Wd вправо, помещает CY в бит 15 Wd, и переносит состояние бита 00 в CY.



**Предосторожности**

Для установки состояния CY используйте команду STC или CLC для очистки CY перед командой сдвига ROR. чтобы быть уверенным, что в CY содержатся правильное значение.

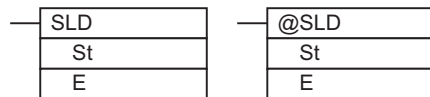
Состояние CY очищается в конце каждого цикла (когда выполняется END).

**Флаги**

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или превышена граница области DM)
- CY:** Получает состояние бита 00
- EQ:** 1, когда содержимое Wd равен нулю, в противном случае = 0.
- N:** 1, когда 1 вдвигается в бит 15 WD.

**5.15.7 SLD - Сдвиг влево на одну цифру**

**Обозначение на ПК**



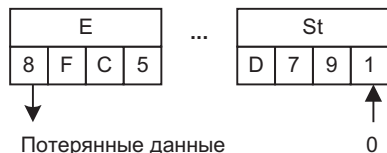
Номер инструкции		Операнды		
Контроллер	№	St	E	
		первое слово	последнее слово	
C200H□-CPU□□-E	74	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM	
C200H□-CPU□□-ZE	074	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, DM, EM	

**Ограничения**

St и E должны быть в одной области данных и ST должно быть меньше либо равно E.

**Описание**

Когда условие исполнения = 0, SLD не выполняется. Когда условие исполнения = 1, SLD сдвигает данные между St и E (включительно) на одну цифру (4 бита) влево. Нули записываются в младшую цифру St, а старшая цифра слова E теряется.



**Предосторожности**

Если во время операции сдвига более 50 слов происходит прерывания питания, операция сдвига может не завершиться.

**Флаги**

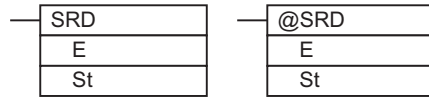
- ER:** Слова St и E лежат в разных областях или St больше E.



Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или превышена граница области DM)

### 5.15.8 SRD - Сдвиг вправо на одну цифру

#### Обозначение на ПКС



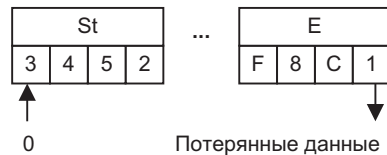
Контроллер	Номер инструкции Nr	Операнды	
		St первое слово	E последнее слово
C200H□-CPU□□-E	75	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	075	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, DM, EM

#### Ограничения

St и E должны быть в одной области данных, и ST должно быть больше либо равно E.

#### Описание

Когда условие выполнения = 0, SRD не выполняется. Когда условие = 1, SRD сдвигает данные между St и E (включительно) на одну цифру (4 бита) вправо. Нуль записывается в старшую цифру слова St, а младшая цифра слова E теряется.



#### Предосторожности

Если во время операции сдвига более 50 слов происходит прерывания питания, операция сдвига может не завершиться. Задавайте диапазон между E и ST не более 50 слов.

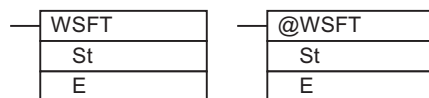
#### Флаги

**ER:** Слова St и E лежат в разных областях или St меньше E.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или превышена граница области DM)

### 5.15.9 WSFT - Сдвиг слова

#### Обозначение на ПКС



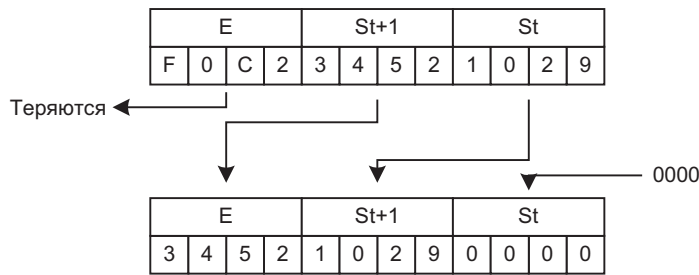
Контроллер	Номер инструкции Nr	Операнды	
		St первое слово	E последнее слово
C200H□-CPU□□-E	16	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	016	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, DM, EM

#### Ограничения

E должно быть больше либо равно St, а St и E должны быть в одной области данных.

#### Описание

Когда условие исполнения = 0, WSFT не выполняется. Когда условие исполнения = 1, WSFT сдвигает данные между St и E словами. Нули записываются в St, а содержание E теряется.



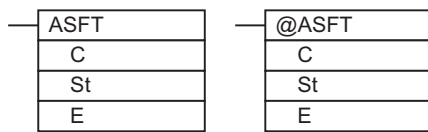
**Флаги**

**ER:** Слова St и E лежат в разных областях или St больше E.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или превышена граница области DM)

**5.15.10 ASFT - Асинхронный регистр сдвига**

**Обозначение на ПКБ**



Номер инструкции		Операнды		
Контроллер	Nr	C	St	E
		слово состояния	первое слово	последнее слово
C200H□-CPU□□-E	17	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	017	IR, SR, HR, AR, LR, DM, EM,#	IR, SR, HR, AR, LR, DM, EM, #	IR, SR, HR, AR, LR, DM, EM, #

**Ограничения**

St и E должны лежать в одной области данных. E должно быть больше либо равно St.

**Описание**

Когда условие исполнения = 0, ASFT действий не производит и программа переходит к следующей команде. Когда условие исполнения = 1, ASFT служит для создания и управления реверсивным асинхронным регистром сдвига на слово между ST и E. Данный регистр сдвигает слова, когда следующее слово в регистре = 0, например, если в регистре нет слов, содержащих нули, ничего не сдвигается. Когда в регистре есть слово, содержащее нули, происходит сдвиг в это слово. Когда содержание слова переместится в следующее слово (содержавшее нули), содержание исходного слова установится в 0. По сути, когда регистр сдвигается, каждое слово нуля в регистре меняется местами со следующим словом. (см. пример).

Направление сдвига (т.е. следующее слово является ближайшим старшим или ближайшим младшим) задается в C. C также служит для сброса регистра. Можно произвести сброс любой области регистра, задав ее St и E.

**Слово управления**

**Биты 00....12**

Не используются

**Бит 13**

Направление сдвига (1 - направление вниз, к младшим словам, 0 - к старшим словам).

**Бит 14**

Бит разрешения сдвига(1 - разрешает регистру работать в соответствии с битом 13, 0 - запрещает).

**Бит 15**

Бит сброса: когда бит 15 = 1, регистр будет сброшен в 0 (между St и E). При нормальной работе бит 15 = 1.

Значение слова состояния	Функция
#4000	Сдвиг вверх (к старшим адресам)
#6000	Сдвиг вниз (к младшим адресам)
#8000	Очистка содержимого ST..E в #0000

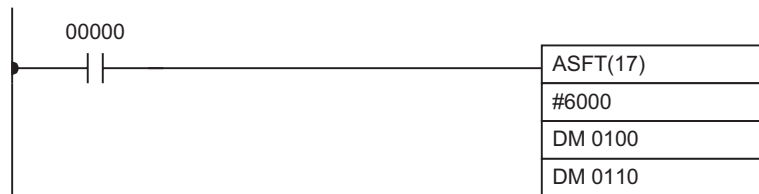
**Флаги**

**ER:** Слова St и E лежат в разных областях или St больше E.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

**Пример**

В примере показано применение команды ASFT для сдвига слов в регистре сдвига из 11 слов между DM 0100..DM 0110 с C = # 6000 (биты 13 и 14 = 1). Показаны также изменения данных и содержимое слова управления.



Адрес	Инструкция	Операнд
00100	LD	00000
00101	ASFT	# 6000
		DM 0100
		DM 0110



## 5.16 Пересылка данных

В данной главе описаны команды, служащие для перемещения данных между различными адресами в областях данных. Данные перемещения можно запрограммировать внутри одной области данных или между разными областями. Команды перемещения данных помогают использовать все области данных ПК. Эффективная связь в системе LINK SYSTEM также требует команд перемещения. Все эти команды изменяют только содержимое слов, куда перемещаются данные, т.е. содержание источника не меняется после исполнения команды.

### 5.16.1 MOV - Пересылка

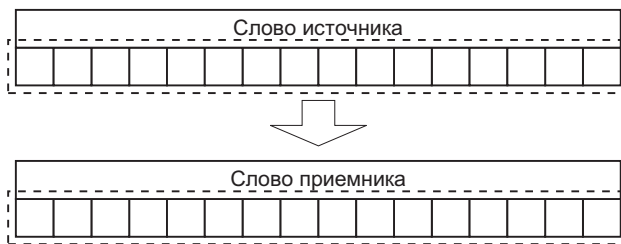
#### Обозначение на ПКС

MOV	@MOV
S	S
D	D

Контроллер	Номер инструкции №	Операнды	
		S слово источник	D слово приемник
C200H□-CPU□□-E	21	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	021	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

#### Описание

Когда условие исполнения = 0, MOV не выполняется. Когда условие исполнения = 1, MOV копирует содержимое S в D.



Состояние битов не изменяется

#### Предосторожности

Номера TC нельзя задавать в качестве D для изменения их текущего значения. Однако текущее состояние легко изменить командой BSET.

#### Флаги

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)
- EQ:** 1, когда в D передаются все нули.
- N:** 1, когда бит 15 D = 1.

### 5.16.2 MVN - Пересылка инверсного значения

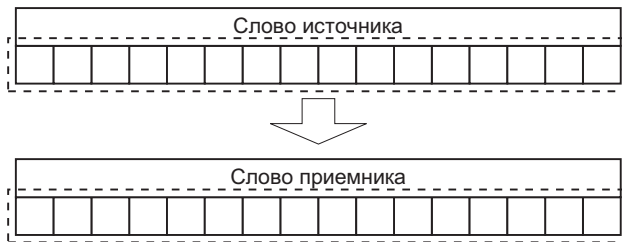
#### Обозначение на ПКС

MVN	@MVN
S	S
D	D

Контроллер	Номер инструкции №	Операнды	
		S слово источник	D слово приемник
C200H□-CPU□□-E	22	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	022	R, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Описание**

Когда условие исполнения = 0, MVN не выполняется. Когда условие исполнения = 1, MVN копирует инвертированное содержимое S (заданное слово или 16-ричная константа из 4 цифр) в D, т.е. для каждого бита S = 1 соответствующий бит в D будет = 0, а для каждого бита S = 0 соответствующий бит в D будет 1.



Состояние битов инвертируется

**Предосторожности**

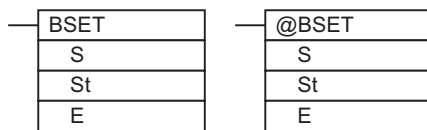
Номера ТС нельзя задавать в качестве D для изменения их текущего значения. Однако текущее состояние легко изменить командой BSET.

**Флаги**

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)
- EQ:** 1, когда в D передаются все нули.
- N:** 1, когда бит 15 D = 1.

**5.16.3 BSET - Заполнение блока**

**Обозначение на ПКС**



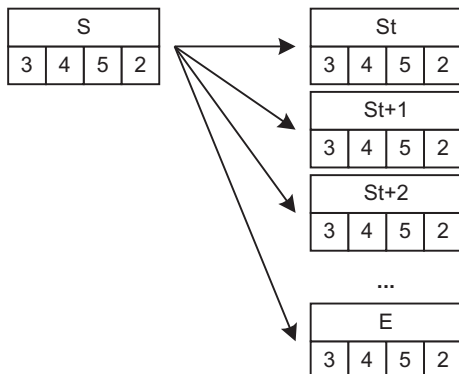
Контроллер	Номер инструкции Nr	Операнды		
		S	St	E
		слово источник1	первое слово	последнее слово
C200H□-CPU□□-E	71	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM
C200H□-CPU□□-ZE	07	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM

**Ограничения**

St должно быть меньше либо равно E, St и E должны лежать в одной области данных.

**Описание**

Когда условие исполнения = 0, BSET не выполняется. Когда условие исполнения = 1, BSET копирует содержимое S во все слова от St до E.



BSET можно использовать для изменения текущего значения таймеров и счетчиков. (Это нельзя сделать командами MOV и MVN). BSET можно также использовать для очистки блока данных, т.е. области DM. для подготовки исполнения других команд.

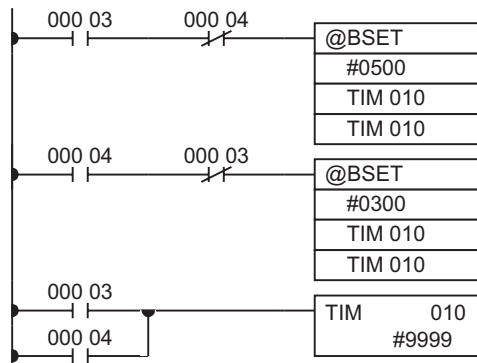
**Флаги**

**ER:** Слова St и E лежат в разных областях или St больше E.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

**Пример**

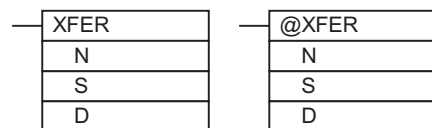
Следующий пример показывает, как использовать BSET для изменения текущего значения таймера в зависимости от состояния IR 00003 и IR 00004. Когда IR 00003 = 1, TIM 010 будет работать в качестве таймера на 50 с. Когда IR 00004 = 1, TIM 010 будет работать как таймер на 30 с.



Адрес	Инструкция	Операнд
00000	LD	000 03
00001	AND NOT	000 04
00002	@BSET	#0500
		TIM 010
		TIM 010
00003	LD	000 04
00004	AND NOT	000 03
00005	@BSET	#0300
		TIM 010
		TIM 010
00006	LD	000 03
00007	OR	000 04
00008	TIM	010
		#9999

**5.16.4 XFER - Пересылка блока**

**Обозначение на РКС**



Контроллер	Номер инструкции №	Операнды		
		N	S	D
C200H□-CPU□□-E	70	число слов (BCD)	первое слово источника	первое слово приемника
		IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM

Номер инструкции		Операнды		
Контроллер	№г	N	S	D
		число слов (BCD)	первое слово источника	первое слово приемника
C200H-CPU-ZE	070	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM

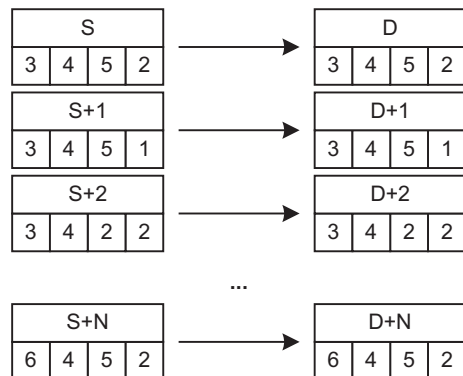
**Ограничения**

S и D должны лежать в одной области данных, но области их блоков могут перекрываться. S и S+N должны лежать в одной области данных, также D и D+N.

N должно быть в двоично-десятичном виде между 0000 и 6144.

**Описание**

Когда условие исполнения = 0, XFER не выполняется. Когда условие исполнения = 1, XFER копирует содержимое S. S+1,..S+N в D, D+1,..D+N.



**Флаги**

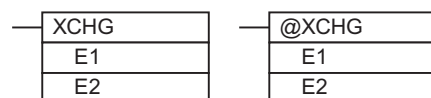
**ER:** N не двоично-десятичное число 0000..2000.

Слова от S до S+N или D до D+N лежат в разных областях.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

**5.16.5 XCHG - Обмен данных**

**Обозначение на РКС**



Номер инструкции		Операнды		
Контроллер	№г	E1	E2	
		слово обмена 1	слово обмена 2	
C200H-CPU-E	73	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	
C200H-CPU-ZE	073	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	

**Описание**

Когда условие исполнения = 0, XCHG не выполняется. Когда условие исполнения = 1, XCHG обменивает содержимое E1 и E2.



Если Вы хотите обменять содержимое блоков с размером больше 1 слова, используйте рабочие слова как промежуточный буфер для хранения одного из блоков и используйте команду XFER три раза.

**Флаги**

**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

## 5.16.6 DIST - Распределение одного слова

## Обозначение на ПКС

DIST	@DIST
S	S
DBs	DBs
C	C

Контроллер	Номер инструкции Nr	Операнды		
		S источник данных	DB базовое слово-премник	C слово состояния
C200H□-CPU□□-E	80	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM, #
C200H□-CPU□□-ZE	080	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM, #

## Ограничения

C должно быть в двоично-десятичном виде. Если C \_ 6655, DBS должно быть в той же области, что и DBS+C. Если C \_ 9000, DBS должно быть в той же области, что и DBS+C-9000.

## Описание

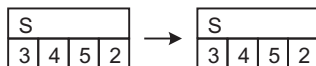
В зависимости от значения C DIST служит либо как команда распределения данных, либо команда для работы со стеком. Если C находится между 0000 и 6655, DIST будет работать как команда распределения данных и копировать содержимое S в DBS+C. Если старшая цифра C = 9, DIST будет работать как команда работы со стеком и создавать стек с числом слов, заданным в трех младших словах C.

## Предосторожности

Стековые операции будут ненадежны, если заданная длина стека отличается от длины, заданной при последнем исполнении DIST или COLL.

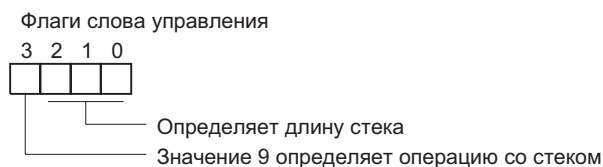
## Операция распределения данных (C=0000..6655)

Когда условие исполнения = 0, DIST не выполняется. Когда условие исполнения = 1, DIST копирует содержимое S в DBS+C, т.е. C добавляется в DBS для определения слова приемника.



## Операция со стеком (C=9000..9999)

Когда условие исполнения = 0, DIST не выполняется. Когда условие исполнения = 1, DIST действует как стек от DBS до DBS+C-9000. DBS - это указатель стека, так что содержимое S копируется в слово, указанное в DBS, и DBS инкрементируется. Флаг N также изменяется.



Данные можно добавлять в стек, пока он не заполнится. DIST как правило, используется совместно с COLL, которая настроена на чтение из стека в режиме FIFO или LIFO. Подробности см. 5.16.7.

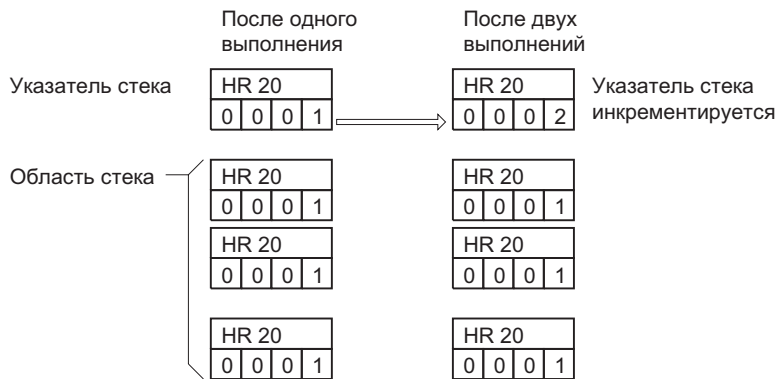
## Пример операции со стеком

В следующем примере содержимое C (LR 10) = 9010, и DIST используется для записи числовых данных #00FF в стек на 10 слов HR 20..HR 29. Во время первого цикла, когда IR 00001 = 1, данные записываются в DBS+1 (HR 21) и указатель стека инкрементируется на 1. Во втором цикле данные записываются в DBS+2 (HR 22) и указатель стека инкрементируется на 1, и так далее.





Адрес	Инструкция	Операнд
00000	LD	000 01
00001	DIST	
		#00FF
		HR 20
		LR 10

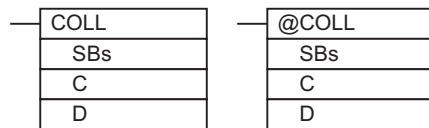


**Флаги**

- ER:** Содержимое C не двоично-десятичное число или 66559000.  
 Когда C\_ 6655, DBS и DBS+C не в одной области данных.  
 Когда C\_ 9000, DBS и DBS+C-9000 не в одной области данных.  
 Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)
- EQ:** 1, когда содержание S = 0; в противном случае = 0.

**5.16.7 COLL - Сбор данных**

**Обозначение на ПК**



Контроллер	Номер инструкции №	Операнды		
		SB базовое слово-источник	C	D слово приемник
C200H□-CPU□□-E	81	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM
C200H□-CPU□□-ZE	081	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM

**Ограничения**

C должно быть в двоично-десятичном виде. Если C\_ 6655, SBS должно быть в той же области, что и SBS+C. Если старшая цифра C = 8 или 9, SBS должно быть в той же области, что и SBS+N (N = 3 младшие цифры C).

**Описание**

В зависимости от содержания управляющего слова C COLL может действовать как команда сбора данных, команда работы со стеком FIFO или команда работы со стеком

LIFO. Если 0000 C 6655, COLL действует как команда сбора данных и копирует содержимое SBS+C в D.

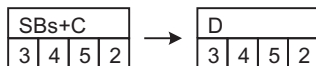
Если старшая цифра C = 9, COLL будет работать как команда работы со стеком FIFO. Если старшая цифра C = 8, COLL будет работать как команда работы со стеком LIFO. Обе операции со стеком используют стек, начиная от SBS с длиной, заданной в 3 младших цифрах C.

**Предосторожности**

Стековые операции будут ненадежны, если заданная длина стека отличается от длины, заданной при последнем исполнении DIST или COLL.

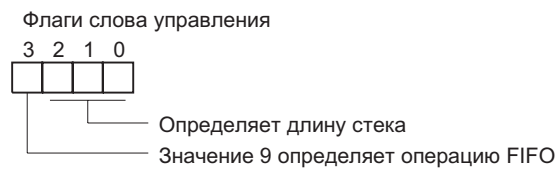
**Операция сбора данных (C=0000..6655)**

Когда условие исполнения = 0, COLL не выполняется. Когда условие исполнения = 1, COLL копирует содержимое SBs + C в D, т.е. C добавляется к SBS для определения слова приемника.



**Операция со стеком FIFO (C=9000..9999)**

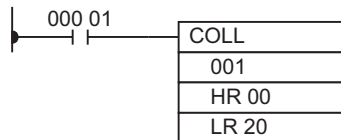
Когда условие исполнения = 0, COLL не выполняется. Когда условие исполнения = 1, COLL копирует данные из слова, записанного в стек раньше всех, в D. Указатель стека, SBS, инкрементируется на 1.



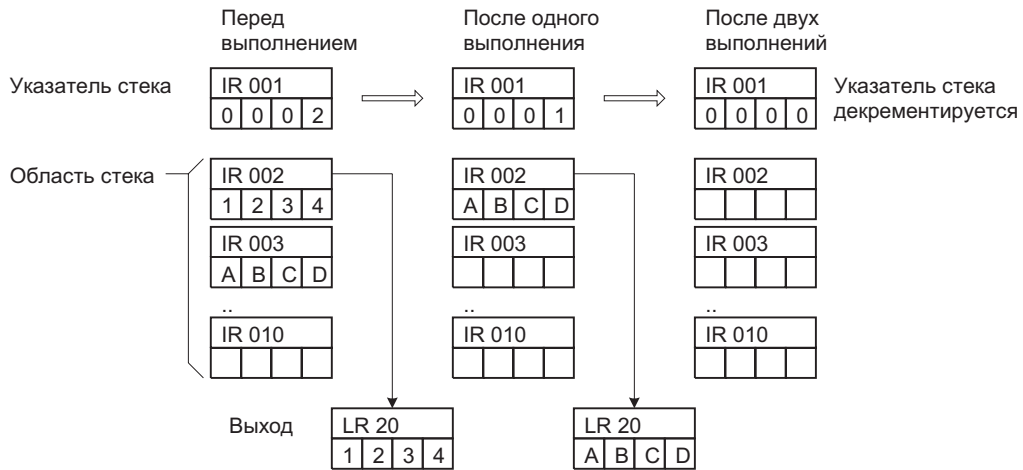
COLL можно использовать совместно с DIST. Подробности см. 5.16.6.

**Пример**

В следующем примере содержимое C (HR 00) = 9010, и COLL используется для копирования слова, записанного в стек (10 слов IR 001..IR 010) раньше всех, в LR 20.



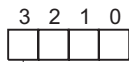
Адрес	Инструкция	Операнд
00000	LD	000 01
00001	COLL	
		001
		HR 00
		LR 20



**Операция со стеком LIFO (C=9000..9999)**

Когда условие исполнения = 0, COLL не выполняется. Когда условие исполнения = 1, COLL копирует данные из слова, записанного в стек последним, в D. Указатель стека, SBS, инкрементируется на 1.

Флаги слова управления

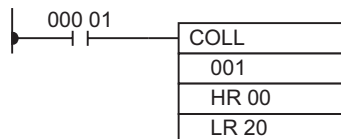


Определяет длину стека  
Значение 8 определяет операцию LIFO

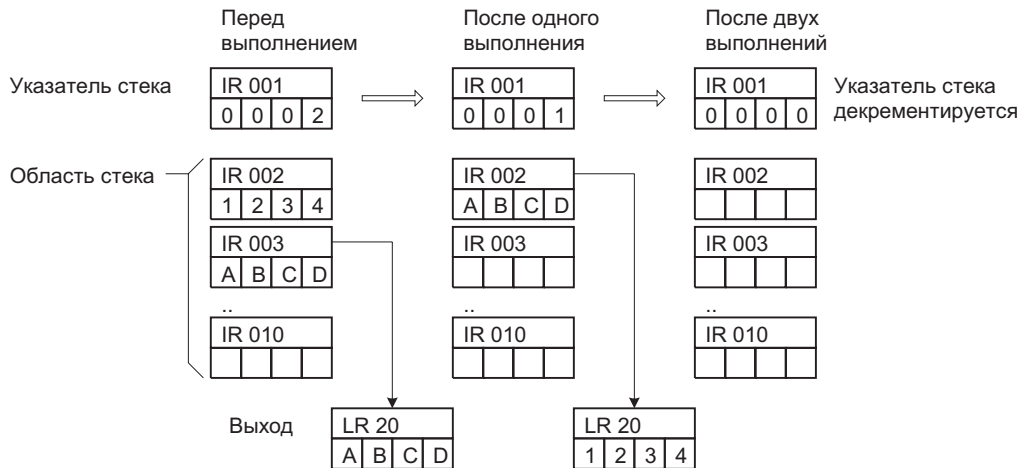
Данные можно добавлять в стек пока он не заполнится. Операции стека DIST, как правило, используется совместно с операцией чтения стека COLL. COLL можно настроить на чтение в режиме FIFO или LIFO. Подробности см. 5.16.6.

**Пример**

В следующем примере содержимое C (HR 00) = 8010, и COLL используется для копирования слова, записанного в стек последним (10 слов IR 001..IR 010), в LR 20.



Адрес	Инструкция	Операнд
00000	LD	000 01
00001	COLL	001
		HR 00
		LR 20



**Флаги**

- ER:** Содержимое C не двоично-десятичное число или 66558000. Когда C\_ 6655, SBS и SBS+C не в одной области данных. Когда C\_ 8000, начало и конец стека не в одной области данных или значение указателя стека превышает длину стека. Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- EQ:** 1, когда передаваемые данные = 0; в противном случае = 0.

**5.16.8 MOVB - Переслать бит**

**Обозначение на ПКС**

MOVB	@MOVB
S	S
Bi	Bi
D	D

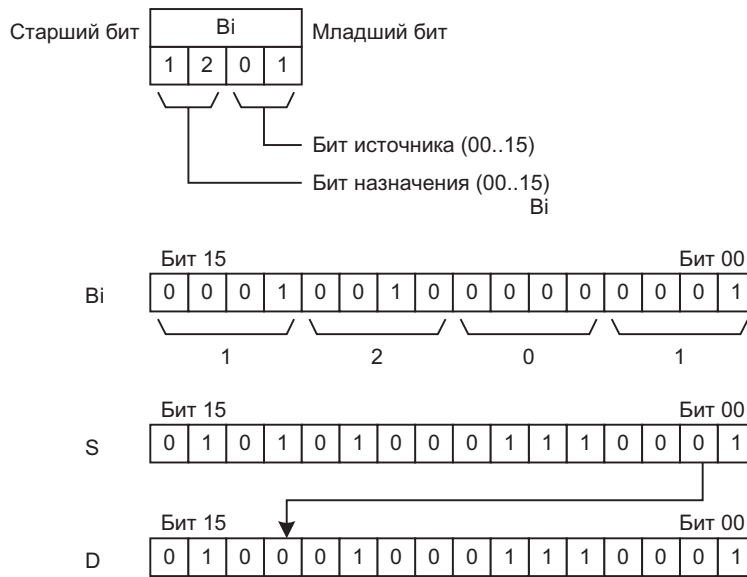
Контроллер	Номер инструкции	Операнды		
		S	Bi	D
C200H□-CPU□□-E	82	слово источник	слово адреса (BCD)	слово приемник
C200H□-CPU□□-ZE	082	IR, SR, HR, AR, LR, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
		IR, SR, HR, AR, LR, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Ограничения**

Две младших цифры и две старших цифры Bi должны быть в диапазоне 00 - 15 каждая.

**Описание**

Когда условие исполнения = 0, MOVB не выполняется. Когда условие исполнения = 1, MOV копирует указанный бит S в указанный бит D. Биты S и D задаются в Bi. Две младшие цифры слова Bi задают бит источника, две старшие цифры слова Bi задают бит приемника.

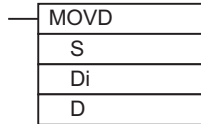


**Флаги**

- ER:** С не в двоично-десятичном виде, или определяет несуществующий бит (т.е. должно быть между 00 и 15).  
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или превышена граница области DM)
- N:** 1, если старший бит содержимого D = 1, в противном случае = 0.

**5.16.9 MOVD - Переслать цифру**

**Обозначение на ПКС**



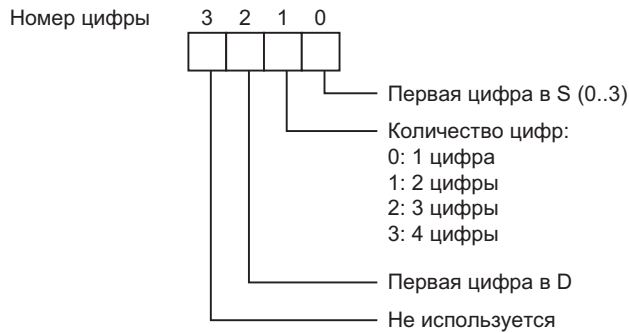
Контроллер	Номер инструкции Nr	Операнды		
		S слово источник	DI слово адреса (BCD)	D слово приемник
C200H□-CPU□□-E	83	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM
C200H□-CPU□□-ZE	083	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM

**Ограничения**

Три младших цифры и две старших цифры DI должны быть в диапазоне 0..3.

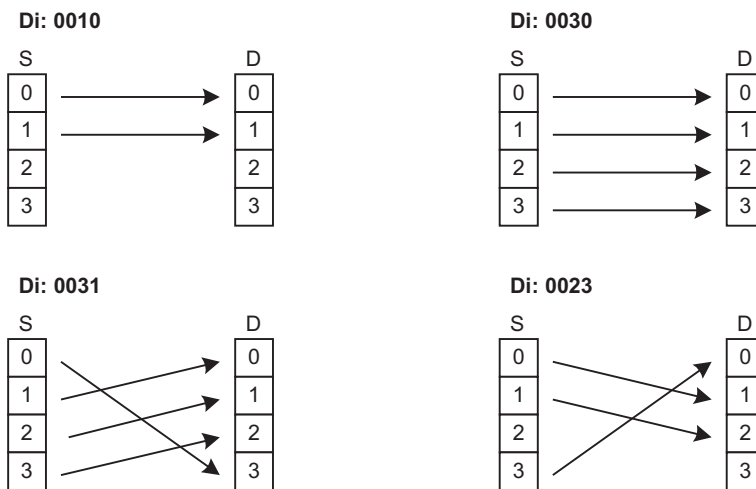
**Описание**

Когда условие исполнения = 0, MOVD не выполняется. Когда условие исполнения = 1, MOVD копирует содержание указанные цифры из S в указанные цифры D. За один раз можно пересылать до 4 цифр. Первая цифра, подлежащая копированию, количество цифр для копирования и первая цифра в приемнике задаются следующим образом. Цифры из S будут копироваться последовательно в D, начиная с указанной цифры. Если последняя цифра в S либо D, дальнейшие операции продолжаются с цифры 0.



**Определитель цифр**

На рисунке показаны примеры перемещения данных для разных значений DI.

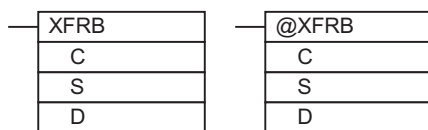


**Флаги**

**ER:** По крайней мере, одна из трех младших цифр DI не в диапазоне 0..3.  
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

**5.16.10 XFRB - Переслать биты**

**Обозначение на РКС**



Номер инструкции		Операнды		
Контроллер	Nr	C	S	D
		слово состояния	первое слово источника	первое слово приемника
C200H□-CPU□□-E	62	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	062	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

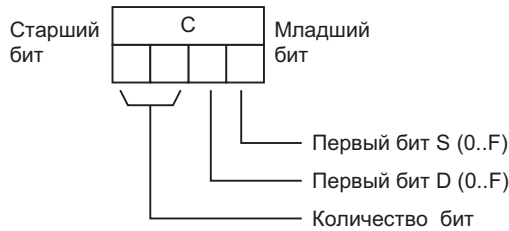
**Ограничения**

Заданные биты источника должны находиться в одной области данных.

Заданные биты приемника должны находиться в одной области данных.

**Описание**

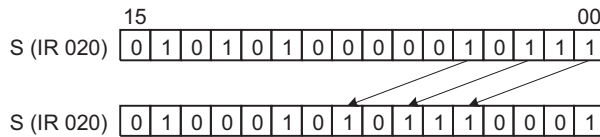
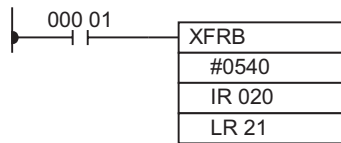
Когда условие исполнения = 0, XFRB не выполняется. Когда условие исполнения = 1, XFRB копирует указанные биты источника в указанные биты приемника. Две младшие цифры слова C задают начальные биты в S и D, а две старшие цифры слова C задают количество битов, подлежащих копированию.



**Замечание** За раз можно копировать до 255 битов.

**Пример**

В следующем примере XFRB используется для передачи 5 битов из IR 020 и IR 021 в LR 00 и LR 01. Стартовый бит в IR 020 = 0, а стартовый бит в LR 21 = 4, так что IR 02000..IR 02004 копируются в LR 2104 - LR 2108.

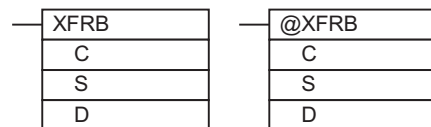


**Флаги**

- ER:** Заданные биты источника находятся в разных областях.  
Заданные биты приемника находятся в разных областях.  
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

**5.16.11 XFR2 - Пересылка блока EM**

**Обозначение на ПКС**



Номер инструкции		Операнды		
Контроллер	№	N	S	D
		число слов (BCD)	первое слово источника	первое слово приемника
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM
C200H□-CPU□□-ZE	062	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM

**Ограничения**

- S и S+N, а также D и D+N должны находиться в одной области данных.
- N должно быть в двоично-десятичном виде.

**Описание**

Когда условие исполнения = 0, XFR2 не выполняется. Когда условие исполнения = 1, XFR2 копирует содержимое S, S+1..S+N в D, D+1.., D+N. Если для S и D используются константы, константа задает адрес в текущем банке EM.

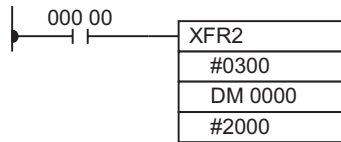
**Флаги**

- ER:** N не в двоично-десятичном виде  
S и S+N, D и D+N находятся в разных областях.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

**Пример**

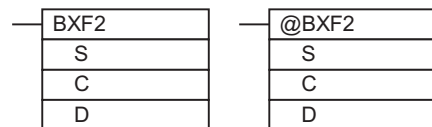
В следующем примере копируется 300 слов из DM 0000..DM 02999 в EM 2000..EM 2299 в текущий банк EM.



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	XFR2	#0300
		DM 0000
		#2000

**5.16.12 BXF2 - Пересылка банка EM**

**Обозначение на РКС**



Контроллер	Номер инструкции	Операнды		
		С	S	D
	№	первое слово состояния	первое слово источника	первое слово приемника
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM
C200H□-CPU□□-ZE	-	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM

**Ограничения**

Значения C+1 должно быть в двоично-десятичном виде от 1 до 6144.

S и S+N, а также D и D+N должны находиться в одной области данных.

N должно быть в двоично-десятичном виде.

**Описание**

Когда условие исполнения = 0, BXF2 не выполняется. Когда условие исполнения = 1, BXF2 копирует содержимое S, S+1..S+N в D, D+1.., D+N. Если для S и D используются константы, константа задает адрес банка EM источника или приемника, заданного в C.

**Слова управления**

В C содержится номера банков источника или приемника, если данные передаются в или из EM. Номера банков игнорируются, если только для S или D не используются константы.

В C+1 содержится число слов для передачи и значение должно быть в двоично-десятичном виде (1..6144).

Слово управления	Биты	Функция
C	0..7	Задаёт номер банка источника (00.. 02)
	8..15	Задаёт номер банка приемника (00.. 02)
C+1	0..15	Задаёт число слов для передачи (1..6144)

**Флаги**

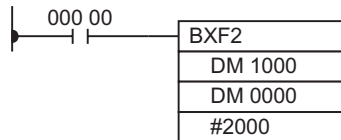
**ER:** N не в двоично-десятичном виде  
S и S+N, D и D+N находятся в разных областях.



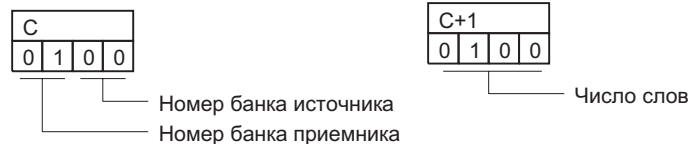
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

**Пример**

В следующем примере копируется 300 слов из DM 0000..DM 02999 в EM 2000..EM 2299 в банк EM 01.

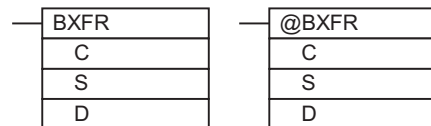


Адрес	Инструкция	Операнд
00000	LD	000 00
00001	BXF2	DM 1000
		DM 0000
		#2000



**5.16.13 BXFR - Пересылка банка EM**

**Обозначение на ПКС**



Контроллер	Номер инструкции Nr	Операнды		
		C	S	D
C200H□-CPU□□-E		первое слово состояния	первое слово источника	первое слово приемника
C200H□-CPU□□-ZE	125	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM

**Замечание** Данной команды у C200H\_-CPU\_-E нет

**Ограничения**

Значения C+1 должно быть в двоично-десятичном виде от 1 до 6144.  
S и S+N, а также D и D+N должны находиться в одной области данных.

**Описание**

Когда условие исполнения = 0, BXFR(125) не выполняется. Когда условие исполнения = 1, BXFR(125) копирует содержимое S, S+1..S+N в D, D+1..., D+N. Можно указать номер банка (в C), если для S и D используется адрес области EM.

**Слова управления**

В C содержится номера банков источника и приемника, если данные передаются в или из EM. Номера банков игнорируются, если только для S или D используются не константы или адреса области EM.

В C+1 содержится число слов для передачи и значение должно быть в двоично-десятичном виде (1..6144).

Слово управления	Биты	Функция
C	0..7	Задаёт номер банка источника (00..0F)
	8..15	Задаёт номер банка приемника (00..0F)
C+1	0..15	Задаёт число слов для передачи (1..6144)

**Флаги**

**ER:** N не в двоично-десятичном виде

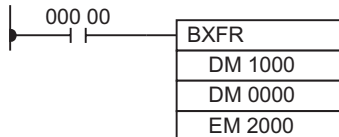
S и S+N, D и D+N находятся в разных областях.

Содержимое слова, в котором находится косвенный адрес DM/EM не в двоично-десятичном виде, или превышена граница области DM/EM.

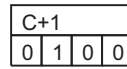
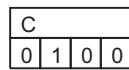
Банк, указанный в C, не существует, когда используется адрес области EM.

**Пример**

В следующем примере копируется содержание 300 слов из DM 0000..DM 2999 в EM 2000..EM 2299 в банк EM 01. (Номер банка источника игнорируется, поскольку адрес области EM не используется).



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	BXFR	DM 1000
		DM 0000
		EM 2000



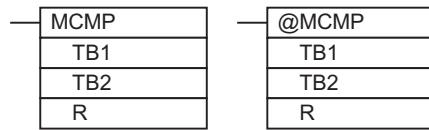
Номер банка источника  
Номер банка приемника

Число слов

## 5.17 Команды сравнения

## 5.17.1 MCMP - Сравнение нескольких слов

## Обозначение на РКС



Контроллер	Номер инструкции Nr	Операнды		
		T1 первое слово таблицы 1	T2 первое слово таблицы 2	R слово результата
C200H□-CPU□□-E	19	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	019	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM

## Ограничения

TB1 и TB1+15, TB2 и TB2+15 должны лежать в одной области данных.

## Описание

Когда условие исполнения = 0, MCMP не выполняется. Когда условие исполнения = 1, MCMP сравнивает содержание TB1 с TB2, TB1+1 с TB2+1,.. TB1+15 с TB2+15. Если первая пара равна, первый бит слова R (бит 00) устанавливается в 0, и так далее, то есть если содержание TB1 равно содержанию TB2, бит 00 устанавливается в 0, если содержание TB1+1 равно содержанию TB2+1, бит 01 устанавливается в 0 и т.д. Остальные биты R будут в состоянии 1.

## Флаги

**ER:** Одна из таблиц (TB..TB15 или TB2..TB2+15) выходит за границу области данных.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

## Пример

В следующем примере показано сравнение и результаты MCMP. Сравнение производится в каждом цикле, когда 00000 = 1.



TB1: IR 100		TB2: DM 0200		R: DM 0300		
IR 100	0100	↔	DM 0200	→	DM 030000	0
IR 101	0200	↔	DM 0201	→	DM 030001	0
IR 102	0210	↔	DM 0202	→	DM 030002	0
IR 103	ABCD	↔	DM 0203	→	DM 030003	1
IR 104	ABCD	↔	DM 0204	→	DM 030004	1
IR 105	ABCD	↔	DM 0205	→	DM 030005	1
IR 106	ABCD	↔	DM 0206	→	DM 030006	1
IR 107	0800	↔	DM 0207	→	DM 030007	0
IR 108	0900	↔	DM 0208	→	DM 030008	0
IR 109	1000	↔	DM 0209	→	DM 030009	0
IR 110	ABCD	↔	DM 0210	→	DM 030010	1
IR 111	ABCD	↔	DM 0211	→	DM 030011	1
IR 112	ABCD	↔	DM 0212	→	DM 030012	1
IR 113	1400	↔	DM 0213	→	DM 030013	0
IR 114	0210	↔	DM 0214	→	DM 030014	0
IR 115	1212	↔	DM 0215	→	DM 030015	1

### 5.17.2 CMP - Сравнение

#### Обозначение на ПКС

CMP
Cp1
Cp2

Контроллер	Номер инструкции Nr	Операнды	
		C1	C2
C200H□-CPU□□-E	20	первое сравниваемое слово IR, SR, HR, AR, LR, TC, DM, #	второе сравниваемое слово IR, SR, HR, AR, LR, TC, DM, #
C200H□-CPU□□-ZE	020	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #

#### Ограничения

При сравнении значений текущего значения таймера или счетчика значение должно быть в двоично-десятичном виде.

#### Описание

Когда условие исполнения = 0, CMP не выполняется. Когда условие исполнения = 1, CMP сравнивает слова Cp1 и Cp2 и выдает результат во флаги GR, EQ и LE в области SR.

#### Предосторожности

Размещение между командой CMP и командами, которые используют флаги GR, EQ и LE, других команд, могут изменить состояние этих флагов. Используйте эти флаги перед тем, как они изменятся.

CMP нельзя использовать для сравнения двоичных чисел со знаком. Вместо этого используйте CPS. Подробности см. 5.17.8.

#### Флаги

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)
- EQ:** 1, если Cp1 = Cp2.
- LE:** 1, если Cp1 меньше Cp2.
- GR:** 1, если Cp1 больше Cp2.

Флаг	Адрес	C1 < C2	C1 = C2	C1 > C2
GR	25505	0	0	1

EQ	25506	0	1	0
LE	25507	1	0	0

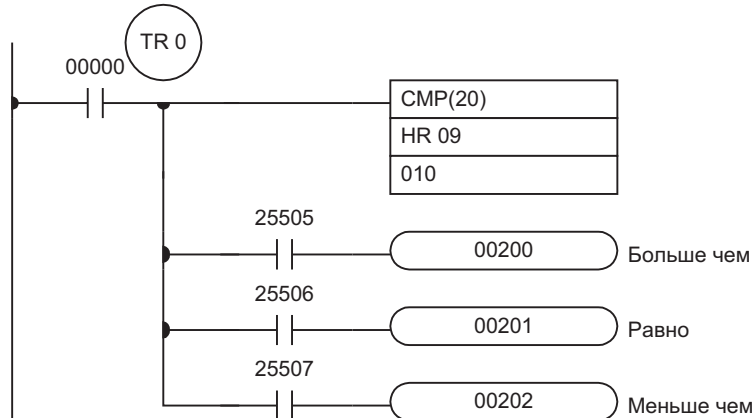
**Пример 1: Сохранение результатов CMP**

В следующем примере показано, как сохранить результаты сравнения немедленно.

Если содержание HR 09 содержания ячейки 10, 00200 устанавливается в 1.

Если содержание HR 09 = содержанию ячейки 10, 00201 устанавливается в 1

Если содержание HR 09 содержания ячейки 10, 00202 устанавливается в 1.

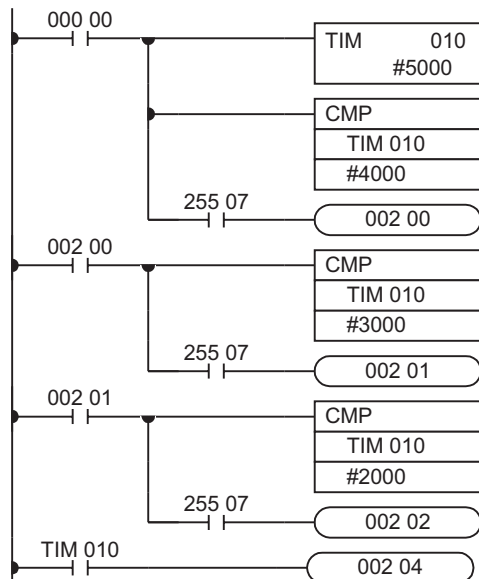


Иногда требуется использование только одного из трех значений, тогда использование TR 0 необязательно. При таком стиле программирования 00200, 00201 и 00202 изменяются только при выполнении CMP.

**Пример 2: Индикация во время операций таймера**

Следующий пример использует TIM, CMP и флаг LE (25507) для выдачи результата в определенное время при уменьшении значения таймера. Таймер стартует при включении 00000 = 1. Когда значение 0000 = 0, TIM 010 сбрасывается, и следующие два сравнения не выполняются (т. е. выполняются с условием выполнения 0). Выход 00200 устанавливается через 100 сек., выход 00201 после 200 сек., выход 00202 после 300 секунд и выход 00204 после 500 с.

Структура ветвления данной схемы важна для того, чтобы обеспечить должное управление 00200, 00201 и 00202 при обратном отсчете таймера. Поскольку все сравнения используют текущее значение таймера, второй операнд каждого CMP должен быть 4-разрядным двоично-десятичным числом.



## 5.17.3 CMPL - Сравнение чисел двойной длины

## Обозначение на РКС

CMPL
Ср1
Ср2
-

Контроллер	Номер инструкции №	Операнды		
		Ср1 первое слово первой сравниваемой пары	Ср2 второе слово второй сравниваемой пары	-
C200H□-CPU□□-E	60	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	
C200H□-CPU□□-ZE	060	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	

## Ограничения

Ср1 и Ср1 + 1 должны лежать в одной области данных

## Описание

Когда условие исполнения = 0, CMPL не выполняется. Когда условие исполнения = 1, CMPL объединяет 4-значное 16-ричное содержание Ср1 с 4-значным содержимым Ср1+1 и содержимое Ср2 с Ср2+1 для создания 8-значных 16-ричных чисел (Ср1+1, Ср1 и Ср2+1, Ср2). Два восьмизначных числа сравниваются и результатом являются флаги GR, EQ, LE в области SR.

## Предосторожности

Размещение между командой CMP и командами, которые используют флаги GR, EQ и LE, других команд, могут изменить состояние этих флагов. Обязательно произведите обращения к этим флагам перед тем, как они изменятся.

CMP нельзя использовать для сравнения двоичных чисел со знаком. Вместо этого используйте CPSL. Подробности см. 5.17.9.

## Флаги

**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

**EQ:** 1, если Ср1+1, Ср1 равно Ср2+1, Ср2.

**LE:** 1, если Ср1+1, Ср1 меньше Ср2+1, Ср2.

**GR:**

1, если Ср1+1, Ср1 больше Ср2+1, Ср2.

## Пример 1: Сохранение результатов CMPL

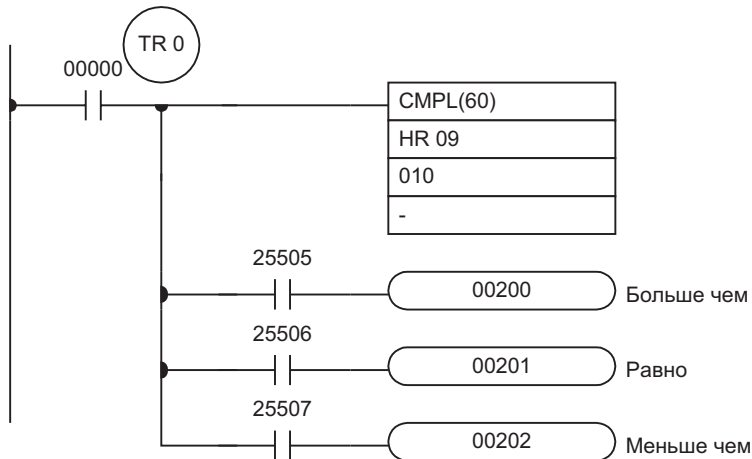
В следующем примере показано, как сохранить результаты сравнения немедленно.

Если содержание HR 10, HR 09 больше содержания ячейки 011, 010, тогда 00200 устанавливается в 1.

Если содержание HR 10, HR 09 равно содержанию ячейки 011, 010, тогда 00201 устанавливается в 1.

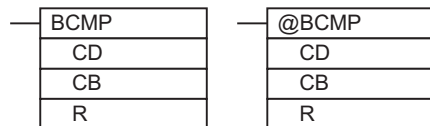
Если содержание HR 10, HR 09 меньше содержания ячейки 011, 010, тогда 00202 устанавливается в 1.

Иногда требуется использование только одного из трех значений, тогда использование TR 0 необязательно. При таком стиле программирования 10200, 10201 и 10202 изменяются только при выполнении CMPL.



### 5.17.4 BCMP - Блочное сравнение

#### Обозначение на ПКС



Контроллер	Номер инструкции №	Операнды		
		CD маска	CB первое слово блока	R результат
C200H□-CPU□□-E	68	IR, SR, HR, AR, LR, TC, DM, #	IR, HR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM
C200H□-CPU□□-ZE	068	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM

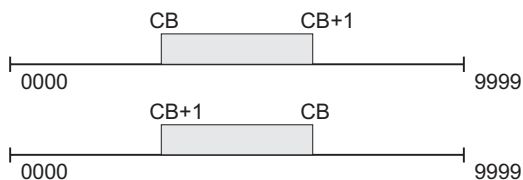
#### Ограничения

Каждый адрес нижней границы должен быть меньше либо равен адресу верхней границы.

#### Описание

Когда условие исполнения = 0, BCMP не выполняется. Когда условие исполнения = 1, BCMP сравнивает CD с зонами, заданными блоком, состоящим из CB, CB+1,.. CB+31. Каждая зона задается двумя словами, первое задает нижнюю границу, второе - верхнюю. Если CD оказывается внутри одной из таких зон, (включая верхнюю и нижнюю границы) устанавливается соответствующий бит в слове R. Прделанное сравнение и. Остальные значения слова R = 0.

Как правило, первое слово в зоне меньше второго, но если первое слово в зоне больше второго, соответствующий бит в R установится в 1, когда CD вне зоны, заданной двумя словами, как показано на следующей диаграмме.

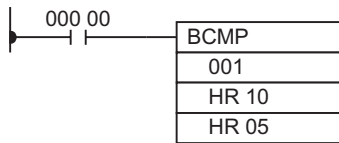


#### Флаги

- ER:** Блок сравнения (т.е. CB.. CB+31) выходит за границы области данных. Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

#### Пример

Следующий пример показывает процесс сравнения и полученные результаты BCMP. Здесь, сравнение происходит каждый цикл, когда IR 0000 = 1.

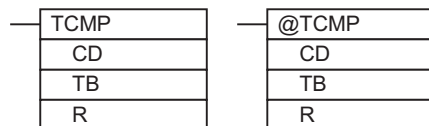


CD: 001    IR 001    0210    Сравнить данные в IR 001 (там содержится 210) с приведенными зонами

Верхние границы		Нижние границы		R: HR 05	
HR 10	0000	HR 11	0100	HR 0500	0
HR 12	0101	HR 13	0200	HR 0501	0
HR 14	0201	HR 15	0300	HR 0502	1
HR 16	0301	HR 17	0400	HR 0503	0
HR 18	0401	HR 19	0500	HR 0504	0
HR 20	0501	HR 21	0600	HR 0505	0
HR 22	0601	HR 23	0700	HR 0506	0
HR 24	0701	HR 25	0800	HR 0507	0
HR 26	0801	HR 27	0900	HR 0508	0
HR 28	0901	HR 29	1000	HR 0509	0
HR 30	1001	HR 31	1100	HR 0510	0
HR 32	1101	HR 33	1200	HR 0511	0
HR 34	1201	HR 35	1300	HR 0512	0
HR 36	1301	HR 37	1400	HR 0513	0
HR 38	1401	HR 39	1500	HR 0514	0
HR 40	1501	HR 41	1600	HR 0515	0

### 5.17.5 TCMP - Табличное сравнение

#### Обозначение на ПКС



Контроллер	Номер инструкции	Операнды			
		№г	CD	ТВ	R
			маска	первое слово таблицы	результат
C200H□-CPU□□-E	85		IR, SR, HR, AR, LR, TC, DM, #	IR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM
C200H□-CPU□□-ZE	085		IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM

#### Ограничения

ТВ и ТВ+15 должны быть в одной области данных.

#### Описание

Когда условие исполнения = 0, TCMP не выполняется. Когда условие исполнения = 1, TCMP сравнивает CD с содержанием ТВ, ТВ+1, ТВ+2,...ТВ+15. Если CD равно содержимому одному из слов, устанавливается соответствующий бит в R, например, если содержание CD = содержанию ТВ, бит 00 установится в 1, если содержание CD = содержанию ТВ+1, бит 01 установится в 1 и т.д. Остальные биты в R будут установлены в 0.

#### Флаги

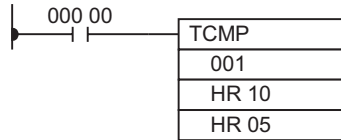
**ER:** Таблица сравнения (т.е. ТВ..ТВ+15) превышает область данных.



Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

**Пример**

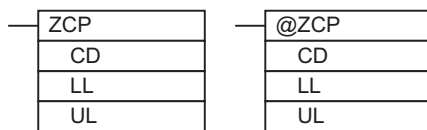
Следующий пример показывает процесс сравнения и полученные результаты TCMP. Здесь, сравнений происходит каждый цикл, когда IR 0000 = 1.



CD: 001	Верхние границы	R: 216
IR 001	DM 0000 0100	IR 21600 0
	DM 0001 0200	IR 21601 0
	DM 0002 0210 →	IR 21602 1
	DM 0003 0400	IR 21603 0
	DM 0004 0500	IR 21604 0
	DM 0005 0600	IR 21605 0
	DM 0006 0210 →	IR 21606 1
	DM 0007 0800	IR 21607 0
	DM 0008 0900	IR 21608 0
	DM 0009 1000	IR 21609 0
	DM 0010 0210 →	IR 21610 1
	DM 0011 1200	IR 21611 0
	DM 0012 1300	IR 21612 0
	DM 0013 1400	IR 21613 0
	DM 0014 0210 →	IR 21614 1
	DM 0015 1600	IR 21615 0

**5.17.6 ZCP - Сравнение с зоной**

**Обозначение на ПКС**



Контроллер	Номер инструкции	Операнды		
		CD	LL	UL
		маска	нижний предел	верхний предел
C200H□-CPU□□-E	88	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #
C200H□-CPU□□-ZE	088	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #

**Ограничения**

LL должна быть меньше либо равна UL.

**Описание**

Когда условие исполнения = 0, ZSP не выполняется. Когда условие исполнения = 1, ZSP сравнивает CD в зоне, заданной нижней границей LL и верхней границей UL и выдает результат во флаги GR, EQ, LE в области SR. Результирующее состояние флагов приведено в таблице.

Результат сравнения	Состояние флага		
	GR (SR 25505)	EQ (SR 25506)	LE (SR 25507)
CD<LL	0	0	1
LL≤CD≤UL	0	1	0
UL<CD	0	0	0

**Предосторожности**

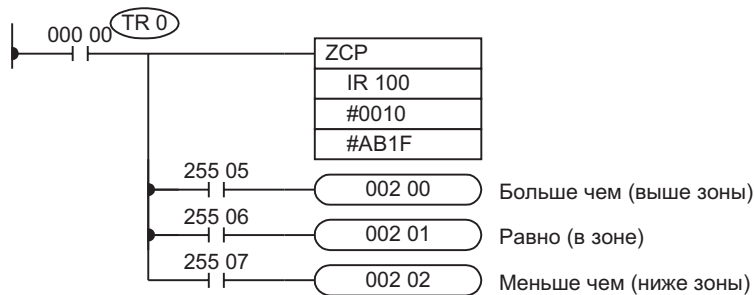
Размещение других команд между командой ZSP и командами, которые используют флаги GR, EQ и LE, может изменить состояние этих флагов. Обязательно произведите обращение к этим флагам перед тем, как они изменятся.

**Флаги**

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)  
LL больше UL.
- EQ:** 1, если LL≤CD≤UL
- LE:** 1, если CD<UL
- GR:** 1, если CD>UL

**Пример: Сохранение результатов ZCP**

В следующем примере показано, как непосредственно сохранить результаты сравнения.



**5.17.7 ZCPL - Сравнение с зоной чисел двойной длины**

**Обозначение на ПКС**

ZCPL
CD
LL
UL

Номер инструкции		Операнды		
Контроллер	Nr	CD маска	LL нижний предел	UL верхний предел
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM
C200H□-CPU□□-ZE	116	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM

**Ограничения**

Значение LL+1, LL (8 цифр) должно быть меньше либо равно UL+1, UL.  
CD и CD+1, LL и LL+1, UL и UL+1 должны быть в одной области данных.

**Описание**

Когда условие исполнения = 0, ZSPL не выполняется. Когда условие исполнения = 1, ZSPL сравнивает значение CD+1, CD (8 цифр) с зоной, заданной нижней границей LL+1, LL и верхней границей UL+1, UL и выдает результат во флаги GR, EQ, LE в области SR. Результирующее состояние флагов приведено в таблице.

Результат сравнения	Состояние флага		
	GR (SR 25505)	EQ (SR 25506)	LE (SR 25507)

CD, CD+1<LL+1,LL	0	0	1
LL+1,LL≤ CD, CD+1≤UL+1,UL	0	1	0
UL+1,UL<CD,CD+1	1	0	0

**Предосторожности**

Размещение других команд между командой ZSPL и командой, которая использует флаги GR, EQ и LE, может изменить состояние этих флагов. Обязательно произведите обращение к этим флагам перед тем, как они изменятся.

**Флаги**

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)  
LL+1,LL больше чем UL+1, UL
- EQ:** 1, если LL+1,LL≤CD, CD+1≤UL+1, UL
- LE:** 1, если CD, CD+1<LL,LL1
- GR:** 1, если CD, CD+1>UL+1,UL

**Пример**

Пример см. 5.17.6. Единственная разница между ZCP и ZCPL - это количество цифр в сравниваемых числах.

**5.17.8 CPS - Сравнение двоичных чисел со знаком**

**Обозначение на РКС**

CPS
Cp1
Cp2
000

Контроллер	Номер инструкции	Операнды			
		№	Cp1	Cp2	000
			первое слово	второе слово	установите в 000
C200H□-CPU□□-E	-		IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	
C200H□-CPU□□-ZE	114		IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	

**Описание**

Когда условие исполнения = 0, CPS не выполняется. Когда условие исполнения = 1, CPS сравнивает 16-битовые (4 цифры) слова Cp1 и Cp2 и выдает результат во флаги GR, EQ, LE в области SR.

- Замечание** 1. Подробности о 16-битовых данных со знаком см. 3.2.
- 2. Подробности о сохранении результатов сравнения см. 5.17.2.

**Предосторожности**

Размещение других команд между командой CPS и командами, которые используют флаги GR, EQ и LE, может изменить состояние этих флагов. Обязательно произведите обращение к этим флагам перед тем, как они изменятся.

**Флаги**

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)
- EQ:** 1, если Cp1 равно Cp2.
- LE:** 1, если Cp1 меньше Cp2.
- GR:** 1, если Cp1 больше Cp2.

Результат сравнения	Состояние флага		
	GR (SR 25505)	EQ (SR 25506)	LE (SR 25507)
Cp1<Cp2	0	0	1

Ср1=C2	0	1	0
Ср1>Ср2	1	0	0

### 5.17.9 CPSL - Сравнение двоичных слов двойной длины со знаком

#### Обозначение на РКС

CPSL
Ср1
Ср2
000

Номер инструкции		Операнды		
Контроллер	№	Ср1 первое слово	Ср2 второе слово	000
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	установите в 000
C200H□-CPU□□-ZE	115	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	

#### Ограничения

СР1 и СР1+1, СР2 и СР2+1 должны быть в одной области данных.

#### Описание

Когда условие исполнения = 0, CPSL не выполняется. Когда условие исполнения = 1, CPSL сравнивает 32-битовые (8 цифр) слова Ср=1, Ср1 и Ср2+1, Ср2 и выдает результат во флаги GR, EQ, LE в области SR.

- Замечание**
1. Подробности о 32-битовых двоичных данных со знаком см. 3.2.
  2. Подробности о сохранении результатов сравнения см. 5.17.2.

#### Предосторожности

Размещение других команд между командой CPSL и командами, которые используют флаги GR, EQ и LE, может изменить состояние этих флагов. Обязательно произведите обращение к этим флагам перед тем, как они изменятся.

#### Флаги

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)
- EQ:** 1, если Ср1+1, Ср1 равно Ср2+1, Ср2.
- LE:** 1, если Ср1+1, Ср1 меньше Ср2+1, Ср2.
- GR:** 1, если Ср1+1, Ср1 больше Ср2+1, Ср2.

Результат сравнения	Состояние флага		
	GR (SR 25505)	EQ (SR 25506)	LE (SR 25507)
Ср1+1, Ср1<02+2, Ср2	0	0	1
Ср1+1, Ср1=Ср2+2, Ср2	0	1	0
Ср1+1, Ср1>Ср2+1, Ср2	1	0	0

### 5.7.10 Команды входного сравнения

#### Обозначение на РКС

Mnem
S1
S2
-

- Замечание** «Mnem» в обозначении заменяется мнемоникой требуемой команды; «code» заменяется функциональным кодом требуемой команды.

**Описание**

Когда условие исполнения = 0, команды входного сравнения не выполняются и программа исполняется до конца командной строки. Когда условие исполнения = 1, команды входного сравнения сравнивают константы и/или содержание заданных слов (значения со знаком или без знака) и создают условие исполнения = 1, когда условия сравнения соблюдаются. Если условия сравнения соблюдаются, остаток командной строки будет пропущен и исполнение переходит на следующую командную строку.

Имеется 24 команды входного сравнения. В них можно использовать различные комбинации символов и опций. Если опции не указаны, сравнение будет производиться с данными типа одно слово без знака.

Символ	Опция (формат данных)	Опция (длина данных)
= (Равно)	S (данные со знаком)	L (данные двойной длины)
<> (Не равно)		
< (Меньше чем)		
≤ (Меньше или равно)		
> (Больше чем)		
≥ (Больше или равно)		

Команды входного сравнения данных без знака (т.е. команды без опции S) могут работать с двоичными или двоично-десятичными данными без знака. Команды входного сравнения данных со знаком (т.е. команды с опцией S) работают с двоичными данными со знаком.

При использовании команд входного сравнения после команды сравнения помещайте на командной строке программы другие команды.

В следующей таблице приведены функциональные коды, мнемоника, названия и функции команд входного сравнения.

Код	Мнемоника	Название	Функция
300	=	Равно	Истина, когда S1=S2
301	= L	Равно для чисел двойной длины	
302	= S	Равно для чисел со знаком	
303	= SL	Равно для чисел со знаком двойной длины	
305	<>	Не равно	Истина, когда S1_S2
306	<> L	Не равно для чисел двойной длины	
307	<> S	Не равно для чисел со знаком	
308	<> SL	Не равно для чисел со знаком двойной длины	
310	<	Истина, когда S1	
311	< L	Меньше чем для чисел двойной длины	
312	< S	Меньше чем для чисел со знаком	
313	< SL	Меньше чем для чисел со знаком двойной длины	
315	≤	Истина, когда S1_S2	
316	≤ L	Меньше чем или равно для чисел двойной длины	
317	≤ S	Меньше чем или равно для чисел со знаком	
318	≤ SL	Меньше чем или равно для чисел со знаком двойной длины	
320	>	Больше чем	Истина, когда S1S2

Код	Мнемоника	Название	Функция
321	> L	Больше чем для чисел двойной длины	Истина, когда S1_S2
322	> S	Больше чем для чисел со знаком	
323	> SL	Больше чем для чисел со знаком двойной длины	
325	≥	Больше чем или равно	
326	≥ L	Больше чем или равно для чисел двойной длины	
327	≥ S	Больше чем или равно для чисел со знаком	
328	≥ SL	Больше чем или равно для чисел со знаком двойной длины	

**Предосторожности**

Команды входного сравнения нельзя использовать в качестве правосторонних (выходных) команд, т.е. между ними и правой шиной должна стоять правосторонняя (выходная) команда.

**Пример**

**< (310)**

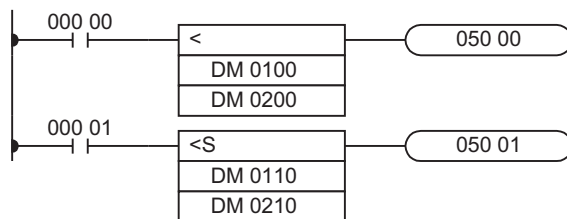
В следующем примере когда IR 00000 = 1, сравнивается содержание DM0100 и DM 0200 (как двоичные значения). Если DM 0100 меньше, чем DM 0200, IR 0500 включается в 1 и исполнение переходит на следующую строку. Если DM 0100 не меньше, чем DM 0200 остаток командной строки пропускается и исполнение переходит на следующую командную строку.

Когда IR 00000 = 0, IR 0500 = 0.

**< S(312)**

В следующем примере когда IR 00001 = 1, сравнивается содержание DM0110 и DM 0210 (как двоичные значения). Если содержимое DM 0110 меньше, чем DM 0210, IR 0501 включается в 1 и исполнение переходит на следующую строку. Если содержимое DM 0110 не меньше, чем DM 0210, остаток командной строки пропускается и исполнение переходит на следующую командную строку.

Когда IR 00001 = 0, IR 0501 = 0.



## 5.18 Команды преобразования данных

Команды преобразования преобразуют данные из одного формата в другой и выдают преобразованные данные в указанные слова результата. Есть преобразования между двоичным (16-ричным видом) и двоично-десятичным в данные для 7-сегментного индикатора, в ASCII. Все эти команды изменяют только содержимое слов приемника, а содержимое источника не изменяется после команд преобразования.

### 5.18.1 BIN - Преобразование двоично-десятичного числа в двоичное

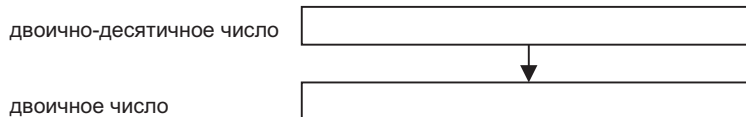
#### Обозначение на ПКС

BIN	@BIN
S	S
R	R

Контроллер	Номер инструкции	Операнды	
		S	R
	№	источник (BCD)	результат
C200H□-CPU□□-E	23	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	023	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

#### Описание

Когда условие исполнения = 0, BIN не выполняется. Когда условие исполнения = 1, BIN преобразует двоично-десятичное содержание S в двоичный эквивалент и помещает двоичное значение в R. Изменяется только содержание R, содержание S не изменяется.



BIN служит для преобразования двоично-десятичного числа в двоичное для того, чтобы числа отображались на программаторе или других программируемых устройствах не в десятичном, а в 16-ричном виде. BIN служит также для преобразования в двоичный вид для выполнения арифметических действий с двоичными, а не с двоично-десятичными данными, Например, когда нужно сложить двоично-десятичное и двоичное число.

#### Флаги

**ER:** Содержание S не в двоично-десятичном виде.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

**EQ:** 1, когда результат равен 0.

**N:** 25402 всегда 0.

### 5.18.2 BINL - Преобразование двоично-десятичного числа двойной длины в двоичное двойной длины

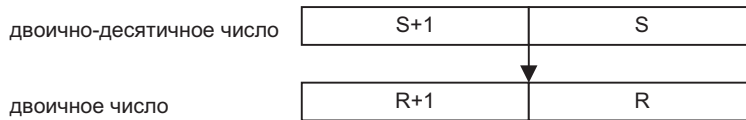
#### Обозначение на ПКС

BINL	@BINL
S	S
R	R

Контроллер	Номер инструкции	Операнды	
		S	R
	№	первое слово источника (BCD)	первое слово результата
C200H□-CPU□□-E	58	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	058	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

**Описание**

Когда условие исполнения = 0, BINL не выполняется. Когда условие исполнения = 1, BINL преобразует длинное число (8 цифр) из S и S+1 в 32-битовое двоичное число и помещает двоичное значение в R, R+1.

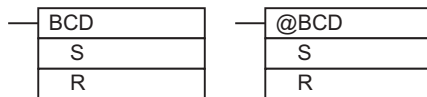


**Флаги**

- ER:** Содержание S и/или S+1 не в двоично-десятичном виде.  
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)
- EQ:** 1, когда результат равен 0.
- N:** 25402 всегда = 0

**5.18.3 BCD - Преобразование из двоичного вида в двоично-десятичный**

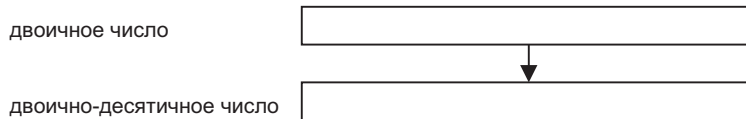
**Обозначение на ПКС**



Контроллер	Номер инструкции №	Операнды		
		S	R	
		источник (двоичное)	результат	
C200H-CPU-E	24	IR, SR, HR, AR, LRC, DM	IR, SR, HR, AR, LR, DM	
C200H-CPU-ZE	024	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, DM, EM	

**Описание**

BCD преобразует двоичное (16-ричное) содержание S в двоично-десятичный эквивалент и помещает двоично-десятичное значение в R. Изменяется только содержание R, содержание S не изменяется.



BCD служит для преобразования двоичного числа в двоично-десятичное для того, чтобы числа отображались на программаторе или других программируемых устройствах не в 16-ричном, а в десятичном виде. BCD служит также для преобразования в двоично-десятичный вид для выполнения арифметических действий с двоично-десятичными, а не с двоичными данными, Например, когда нужно сложить двоично-десятичное и двоичное число.

**Замечание** Если содержимое S превышает 270F, преобразованный результат будет превышать 9999 и BCD не выполнится. Когда команда не выполнялась, содержимое R остается неизменным.

**Двоичные данные со знаком**

BCD нельзя использовать для преобразования двоичных данных со знаком прямо в двоично-десятичные. Для преобразования двоичных данных со знаком сначала определите, преобразуемое значение отрицательное или положительное. Если оно положительно, данную команду можно использовать для преобразования в двоично-десятичное значение. Если отрицательные, используйте команду дополнения до 2 - NEG для превращения в двоичное значение без знака перед выполнением BCD. Подробности о двоичных данных со знаком см. 3.2.

**Флаги**

- ER:** S больше 270F.



Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

**EQ:** 1, когда результат равен 0.

### 5.18.4 BCDL - Преобразование двоичного числа двойной длины в двоично-десятичное число двойной длины

#### Обозначение на ПКС

BCDL	@BCDL
S	S
R	R

Номер инструкции		Операнды		
Контроллер	№	S	R	
		источник (двоичное)	результат	
C200H-CPU□□-E	59	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM	
C200H-CPU□□-ZE	059	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, DM, EM	

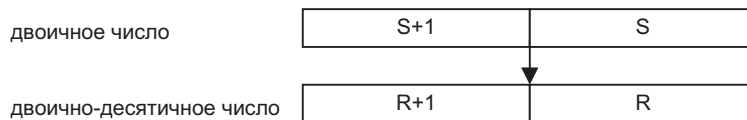
#### Ограничения

Если содержимое S превышает 05F5 E0FF, результат будет больше 9999 9999 и BCDL не выполнится. Когда команда не выполняется, содержание R и R+1 не изменяется.

S и S+1, как и R и R+1 должны лежать в одной области данных.

#### Описание

BCDL преобразует двоичное 32-битовое содержимое S и S+1 в 8 двоично-десятичных цифр и помещает результат преобразования в R и R+1.



#### Двоичные данные со знаком

BCDL нельзя использовать для преобразования двоичных данных со знаком прямо в двоично-десятичные. Для преобразования двоичных данных со знаком сначала определите, преобразуемое значение отрицательное или положительное. Если оно положительно, команду BCD можно использовать для преобразования в двоично-десятичное значение. Если оно отрицательно, используйте команду дополнения до 2 - NEGL для превращения в двоичное значение без знака перед выполнением BCD. Подробности о двоичных данных со знаком см. 3.2.

#### Флаги

**ER:** Содержание R и R+1 превышает 9999 9999

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

**EQ:** 1, когда результат равен 0.

### 5.18.5 SEC - Преобразование Часы-в-секунды

#### Обозначение на ПКС

SEC	@SEC
S	S
R	R
000	000

Номер инструкции		Операнды		
Контроллер	№	S	R	000
		источник (BCD)	результат (BCD)	установите в 000
C200H-CPU□□-E	65	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	
C200H-CPU□□-ZE	065	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	

**Ограничения**

- S и S+1 должны находиться в одной области данных.
- R и R+1 должны находиться в одной области данных.
- S и S+1 должны быть в двоично-десятичном виде и в требуемом формате часы/минуты/секунды.

**Описание**

SEC служит для преобразования из формата часы/минуты/секунды в секунды.  
 В источнике секунды расположены в битах 00..07, минуты в битах 08..15 слова S. Часы расположены в S+1. Максимум 9 999 часов, 59 минут, 59 секунд.  
 Результат выдается в R и R+1. Максимальное значение - 35 999 999 секунд.

**Флаги**

- ER:** S и S+1 или R и R+1 расположены в разных областях памяти.  
 S и S+1 не в двоично-десятичном виде.  
 Количество секунд или минут превышает 59.  
 Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- EQ:** 1, когда результат равен 0000.

**Пример**

Когда 00000 = 0 (т.е. условие исполнения = 1) данная команда преобразует часы/минуты/секунды из HR 12 и HR 13 в секунды и заносит результат в DM 0100 и DM 0101 как показано.



HR 12	3	2	0	7
HR 13	2	8	1	5

2815 часа, 32 минуты,  
0.7 секунды

DM 0100	5	9	2	7
DM 0101	1	0	1	3

10135927 секунд

**5.18.6 HMS - Преобразование Секунды-в-часы**

**Обозначение на ПКС**

HMS	@HMS
S	S
R	R
000	000

Контроллер	Номер инструкции	Операнды		
		S	R	000
		источник (BCD)	результат (BCD)	установите в 000
C200H□-CPU□□-E	66	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	
C200H□-CPU□□-ZE	066	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	

**Ограничения**

- S и S+1 должны находиться в одной области данных.
- R и R+1 должны находиться в одной области данных.
- S и S+1 должны быть в двоично-десятичном виде и находиться между 0 и 35 999 999.

**Описание**

HMS служит для преобразования из формата секунд в формат часы/минуты/секунды.

Количество секунд, находящиеся в S и S1, преобразуется в формат часы/минуты/секунды и результат выдается в R и R+1.

В слове результата секунды расположены в битах 00 - 07, минуты в битах 08 - 15 слова R. Часы расположены в R+1. Максимум 9 999 часов, 59 минут, 59 секунд.

**Флаги**

**ER:** S и S+1 или R и R+1 расположены в разных областях памяти.

S и/или S+1 содержат данные не в двоично-десятичном виде или содержит более 36 000 000 секунд.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

**EQ:** 1, когда результат равен нулю.

**Пример**

Когда 00000 = 0 (т.е. условие исполнения = 1) показанная команда преобразует секунды из слов HR 12 и HR 13 в формат часы/минуты/секунды и заносит результат в DM 0100 и DM 0101.



HR 12	5	9	2	7
HR 13	1	0	1	3

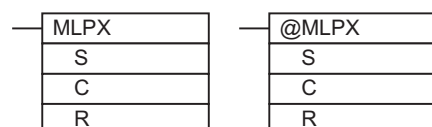
10135927 секунд

DM 0100	3	2	0	7
DM 0101	2	8	1	5

2815 часа, 32 минуты,  
0.7 секунды

**5.18.7 MLPX - Преобразовать 4-в-16 / 8-в-256**

**Обозначение на ПКС**



Контроллер	Номер инструкции №	Операнды		
		S источник	C слово состояния	R первое слово результата
C200H□-CPU□□-E	76	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	076	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Ограничения**

Когда старшая цифра C = 0, две правые цифры C каждая должна быть 0..3.

Когда старшая цифра C = 1, две правые цифры C каждая должна быть 0..1.

Слова результата должны находиться в одной области памяти.

**Описание**

В зависимости от значений C MLPX работает, как преобразователь 4 бит в 16 бит, или как 8-бит в 256 бит.

**Преобразователь 4-бит в 16-бит**

MLPX работает как преобразователь 4-бит в 16-бит, когда старшая цифра C=0.

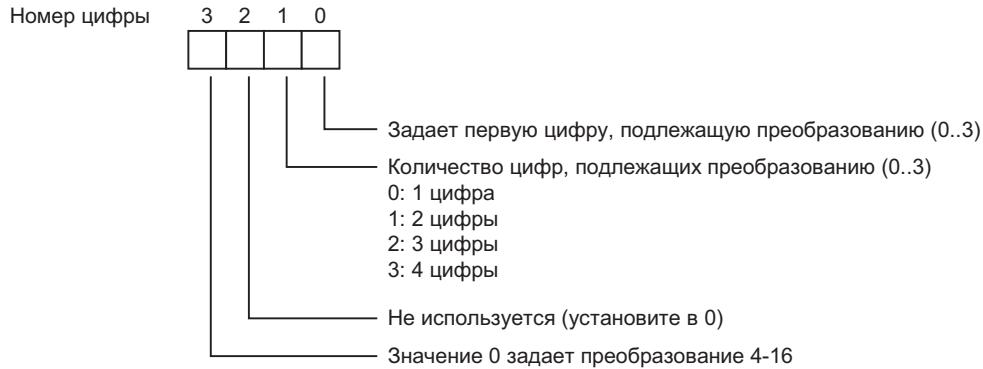
16-ричное значение цифр в источнике S служит для указания битов в словах результата

(до четырех). Указанный бит в каждом слове результата будет = 1, а остальные 15 бит в каждом слове = 0.

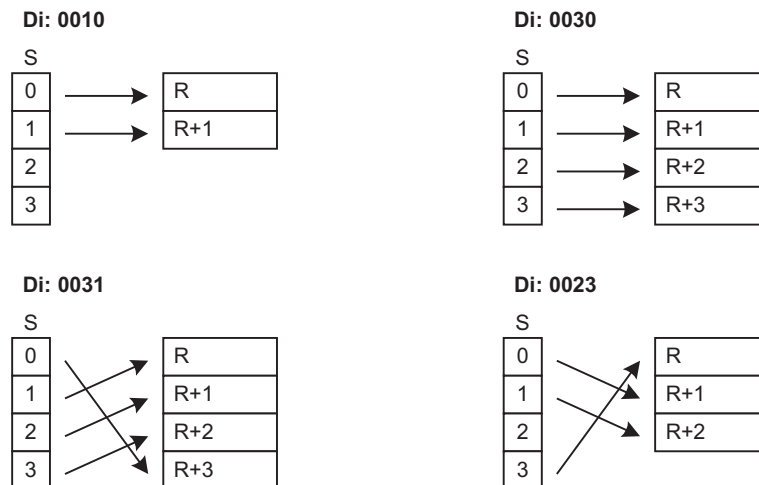
Когда условие исполнения = 0, MLPX не выполняется. Когда условие исполнения = 1, MLPX преобразует до четырех 4-х битовых 16-ричных цифр из S в десятичные значения 0 - 15, каждая из них используется для указания позиции бита. Бит, номер которого соответствует каждому преобразованному значению в слове результата, устанавливается в 1. Если задано более одной цифры, тогда один бит будет установлен в 1 в каждом из последовательных слов, начинающихся с R. (см. пример ниже).

**Слово управления**

Цифры в C расположены следующим образом. Для задания декодирования 4-бит в 16-бит задайте левую цифру C = 0.



На рисунке приведены примеры значения C и преобразования цифра-в-слово, которые производится при данных значениях.



На рисунке приведен пример операции декодирования одной цифры (цифры номер 1 со слове S, т.е. здесь C = 0001).



Первая цифра и количество цифр для преобразования указаны в C. Если задано больше цифр, чем остается в S (считая от первой указанной цифры), остальные цифры будут браться с начала S. Последнее слово, требуемое для загрузки преобразованного результата (R плюс количество цифр, подлежащих преобразованию) должен быть в той же области, что и R, т.е. если преобразуются две цифры, адрес последнего слова

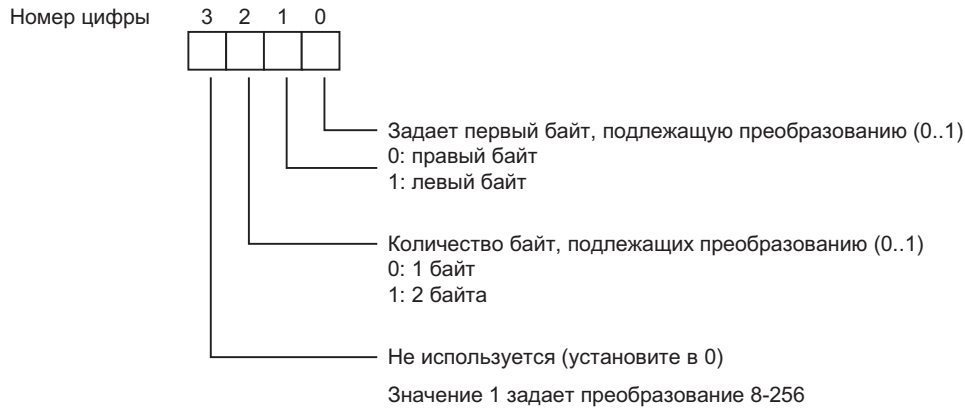
области памяти нельзя назначать. Если преобразуются три цифры, нельзя назначать два последних слова.

**Преобразователь 8-бит в 256-бит**

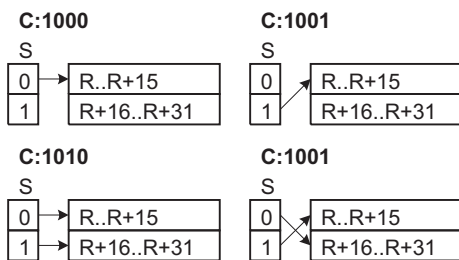
MPLX работает как преобразователь 8-бит в 256-бит, когда старшая цифра C=1. 16-ричное значение двух байт в источнике S служит для указания бита в словах результата (до двух) или двух групп из 16 последовательных слов результата (256 бит). Указанный бит в каждой группе будет = 1, а остальные 255 бит этой группы = 0.

**Слово управления**

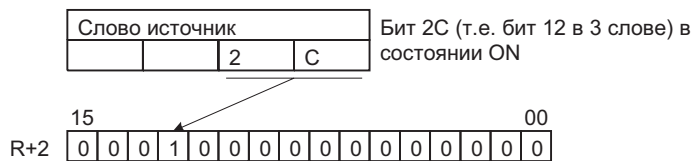
Цифры в C расположены следующим образом. Для задания декодирования 8-бит в 256-бит задайте левую цифру C = 1.



На рисунке приведены 4 возможные значения C и преобразования, которые производится при данных значениях (в S нуль указывает на правый байт, 1 на левый байт).



На рисунке приведен пример операции декодирования одного байта из правого (младшего) байта S (в данном случае C = 1000).

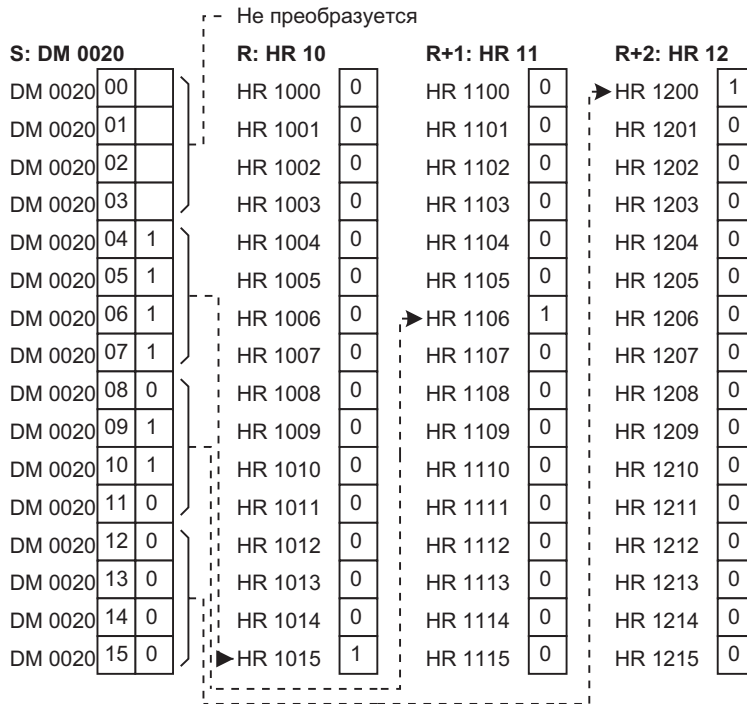


**Флаги**

- ER:** Неопределенное слово управления  
Слова результата находятся не в одной области.  
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

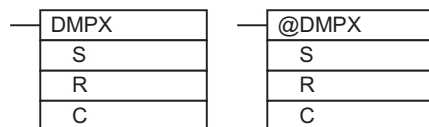
**Пример: Преобразование 4-бит в 16-бит**

Следующая программа преобразует три цифры из LR 20 в позицию битов и устанавливает в 1 соответствующие биты в трех последовательных словах, начинающихся с HR 10.



### 5.18.8 DMPX - Преобразовать 16-в-4 / 256-в-8

#### Обозначение на ПКС



Номер инструкции		Операнды		
Контроллер	№г	S	R	C
		первое слово источника	результат	слово состояния
C200H□-CPU□□-E	77	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, TC, DM, #
C200H□-CPU□□-ZE	077	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM, #

#### Ограничения

Когда старшая цифра C = 0, две правые цифры C каждая должна быть 0..3.

Когда старшая цифра C = 1, две правые цифры C каждая должна быть 0..1.

Все слова результата должны находиться в одной области данных.

#### Описание

В зависимости от значений C DMPX работает как преобразователь 16-бит в 4-бит, или как 256-бит в 8-бит.

#### Преобразователь 16-бит в 4-бит

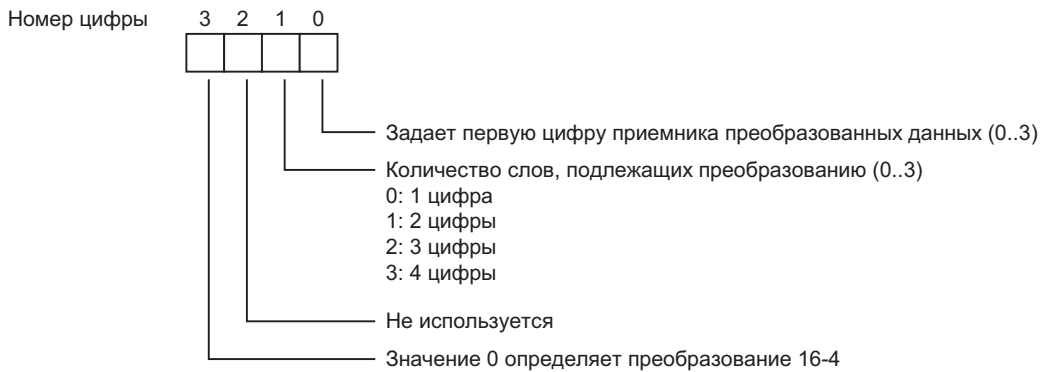
DMPX работает как преобразователь 16-бит в 4-бит, когда старшая цифра C=0.

Когда условие исполнения = 0, DMPX не выполняется. Когда условие исполнения = 1, DMPX определяет позицию старшего, который установлен в 1, слова в S и кодирует его

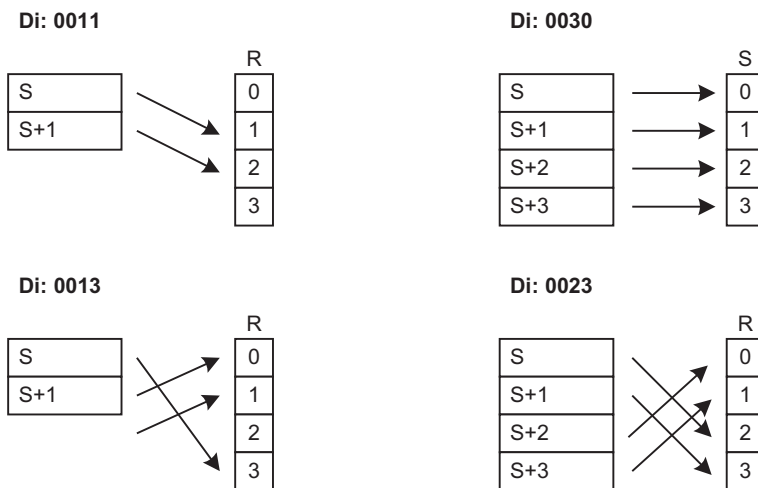
в одну 16-ричную цифру, соответствующую номеру старшего бита, установленного в 1, затем передает 16-ричное значение в указанную цифру в R. Цифры, которым передаются результат, задаются в C (C также определяет количество преобразуемых цифр).

**Слово управления**

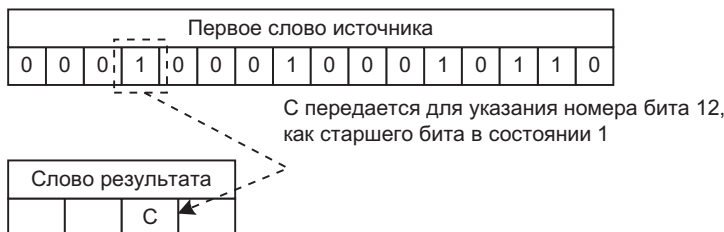
Цифры в C расположены следующим образом. Для задания декодирования 16-бит в 4-бит задайте левую цифру C = 0.



На рисунке приведены примеры значения C и преобразования слов-в-цифру, которые производится при данных значениях.



На рисунке приведен пример операции декодирования одной цифры в цифру номер 1 слова R, (т.е. здесь C = 0001).



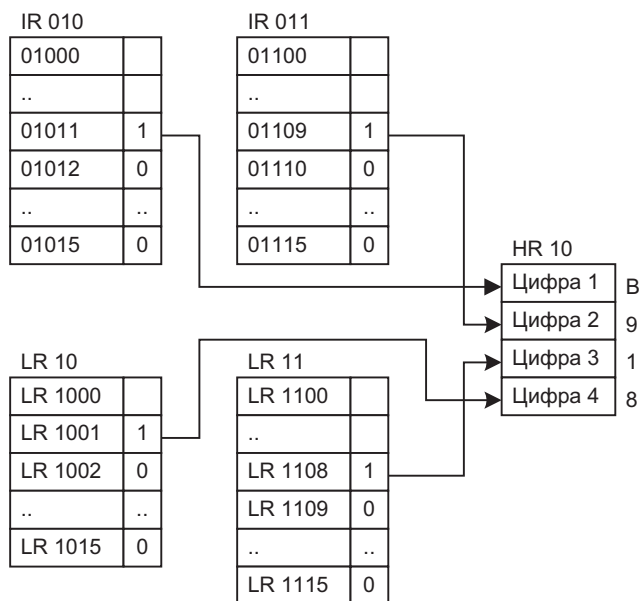
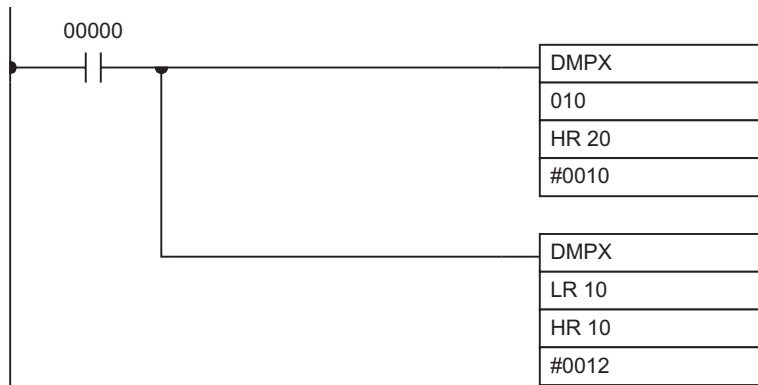
Можно преобразовать до 4 цифр из 4 последовательных слов, начиная с S, и записать их в R в порядке от указанной первой цифры. Если задано больше цифр, чем остается в R (считая от первой заданной цифры), остальные цифры будут помещаться с начала R. Последнее слово, подлежащее преобразованию, (S плюс количество цифр, подлежащих преобразованию) должно быть в той же области, что и S).

**Преобразователь 256-бит в 8-бит**

DMPX работает как преобразователь 266-бит в 8-бит, когда старшая цифра C=1.

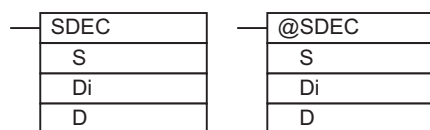






### 5.18.9 SDEC - Преобразование в коды 7-сегментного индикатора

#### Обозначение на ПКС



Контроллер	Номер инструкции	Операнды			
		№	S	DI	D
C200H□-CPU□□-E	78		IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	078		IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM, #

#### Ограничения

Значения DI должны быть в далее приведенном диапазоне.

Все слова приемника должны находиться в одной области памяти.

#### Описание

Когда условие исполнения = 0, SDEC не выполняется. Когда условие исполнения = 1, SDEC преобразует указанные цифры (цифру) слова S в 8-битовый эквивалент, код 7-сегментного дисплея и помещает его в слово (слова) приемника, начинающиеся с D.

Любая или все цифры в S преобразуются последовательно, начиная от заданной цифры. Первая цифра, количество цифр, подлежащих преобразованию, и половина D, принимающая первый код 7-сегментного дисплея (старшие или младшие 8 бит) указаны

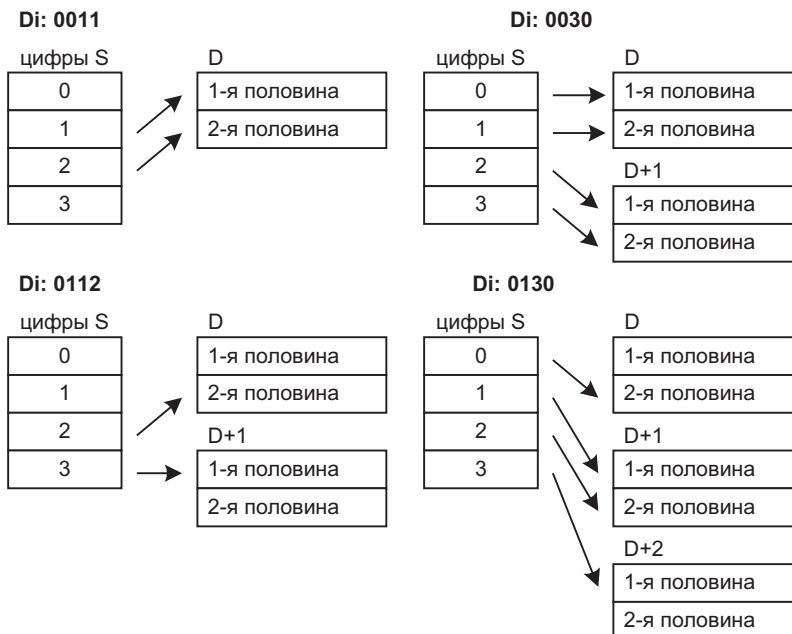
в DI. Если заданы несколько цифр, они должны быть помещены в порядке, начиная с указанной половины D, каждая требует 2 цифры. Если задано больше цифр, чем остается в S (считая от первой заданной цифры), остальные цифры будут браться с начала S.

**Описатель цифр**

Цифры в DI расположены следующим образом.

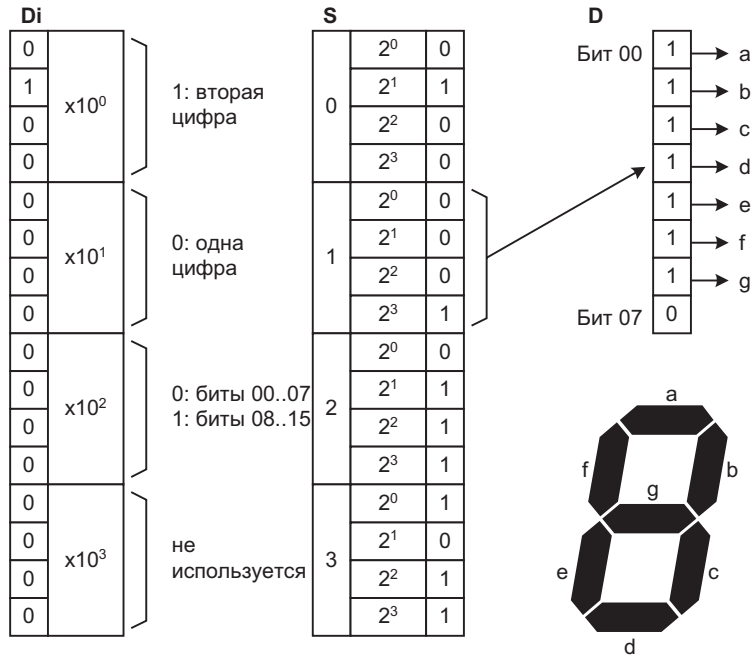


На рисунке приведены примеры значения DI и результаты преобразования 4-битовых двоичных значений в коды 7-сегментного индикатора, которые производится при данных значениях.



**Пример**

В следующие примере показаны данные для высвечивания числа 8. Строчные буквы показывают, какие биты соответствуют сегментам 7-сегментного индикатора. В таблице показаны исходные данные и преобразованный код для всех 16-ричных цифр.



Исходные данные					Преобразованные коды (сегменты)									Индикация
Цифра	Биты				-	g	f	e	d	c	b	a		
0	0	0	0	0	0	0	1	1	0	0	0	0	0	
1	0	0	0	1	0	0	0	1	0	0	0	0	1	
2	0	0	1	0	0	0	0	1	0	0	1	1	2	
3	0	0	1	1	0	0	0	1	0	0	1	1	3	
4	0	1	0	0	0	0	1	1	0	1	0	0	4	
5	0	1	0	1	0	0	1	1	0	1	0	1	5	
6	0	1	1	0	0	0	1	1	0	1	0	1	6	
7	0	1	1	1	0	0	1	1	0	1	1	1	7	
8	1	0	0	0	0	0	1	1	1	0	0	0	8	
9	1	0	0	1	0	0	1	1	1	0	0	1	9	
A	1	0	1	0	0	1	0	0	0	0	0	1	A	
B	1	0	1	1	0	1	0	0	0	1	1	0	B	
C	1	1	0	0	0	1	0	0	0	1	1	1	C	
D	1	1	0	1	0	1	1	0	0	0	0	0	D	
E	1	1	1	0	0	1	1	0	0	0	0	1	E	
F	1	1	1	1	0	1	1	0	0	1	1	0	F	

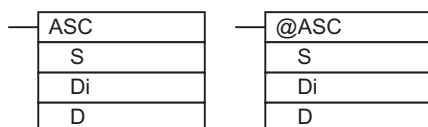
**Флаги**

**ER:** Некорректный определитель цифр или адрес приемника выходит за границы области данных.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

**5.18.10 ASC - Преобразование в коды ASCII**

**Обозначение на ПКС**



Контроллер	Номер инструкции	Операнды		
		S	DI	D
	№г	источник		
C200H□-CPU□□-E	86	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	086	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Ограничения**

Значения DI должны быть в далее приведенном диапазоне.

Все слова приемника должны находиться в одной области данных.

**Описание**

Когда условие исполнения = 0, ASC не выполняется. Когда условие исполнения = 1, ASC преобразует указанные цифры (цифру) слова S в 8-битовый эквивалент кода ASCII и помещает его в слово (слова) приемника, начинающиеся с D.

Любая или все цифры в S преобразуются последовательно, начиная от указанной первой цифры. Первая цифра, количество цифр, подлежащих преобразованию, и половина D, принимающая первый код ASCII (старшие или младшие 8 бит) указаны в DI. Если заданы несколько цифр, они должны быть помещены в порядке, начиная с указанной половины D, каждая требует 2 цифры. Если задано больше цифр, чем остается в S (считая от заданной первой цифры), остальные цифры будут отсчитываться с начала S.

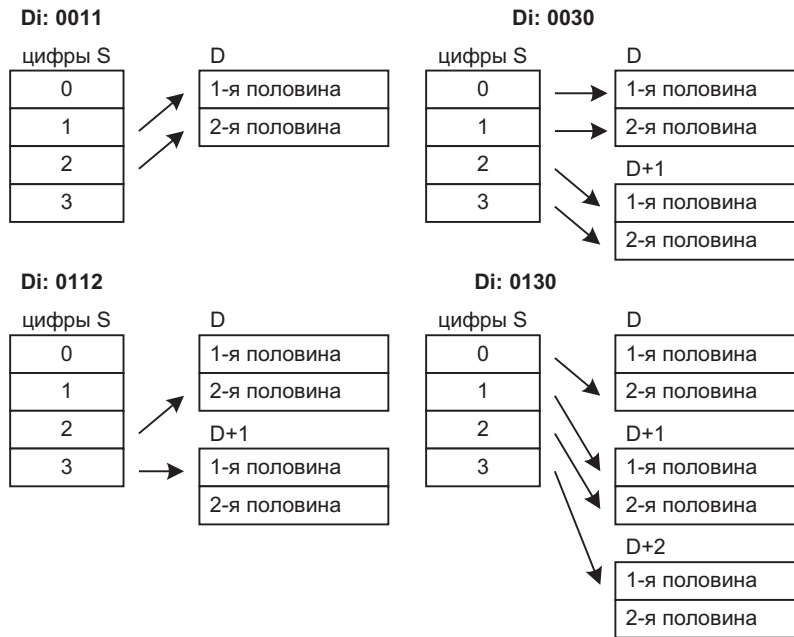
Расширенная таблица символов ASCII приведена в Приложении G.

**Описатель цифр**

Цифры в DI расположены следующим образом.



На рисунке приведены примеры значения DI и результаты преобразования 4-битовых двоичных значений в 8-битовые коды, которые производится при данных значениях.



**Контроль четности**

Старший бит каждого знака ASCII (2 цифры) может быть автоматически настроен на контроль четности или нечетности. Если контроль по четности не задан, старший бит всегда = 0.

Когда контроль четности установлен на четно, старший бит будет принимать такое значение, чтобы общее количество битов, находящихся в состоянии 1, было четным, например, код ASCII "31" (00110001) будет равным "B1" (10110001): бит четности установится в 1 для того, чтобы общее количество битов в состоянии 1 было четным. Код ASCII "36" (00110110) будет равным "36" (00110110): бит четности установится в 0, потому что общее количество битов в состоянии 1 уже четно.

Когда контроль четности установлен на нечетно, старший бит будет принимать такое значение, чтобы общее количество битов, находящихся в состоянии 1, было нечетным.

**Флаги**

**ER:** Некорректный определитель цифр или адрес приемника выходит за границы области данных.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

**5.18.11 HEX - Преобразование из ASCII в 16-ричное число**

**Обозначение на ПКС**

HEX	@HEX
S	S
Di	Di
D	D

Контроллер	Номер инструкции	Операнды		
		S	DI	D
		первое словоисточника		
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	162	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Ограничения**

Значения DI должны быть в далее приведенном диапазоне.

Все слова источника должны находиться в одной области данных.

Байты слов источника должны содержать 16-ричные эквиваленты кодов ASCII, т.е. 30 - 39 (0-9) или 41-46 (A-F).

**Описание**

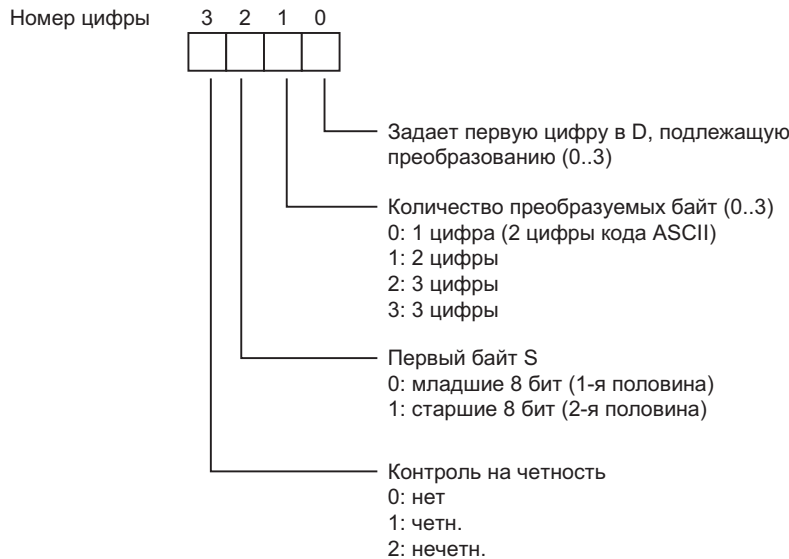
Когда условие исполнения = 0, HEX не выполняется. Когда условие исполнения = 1, HEX преобразует указанные бит (биты) кода ASCII из слова (слов) источника в 16-ричный эквивалент и помещает их в D.

Можно преобразовать до 4 кодов ASCII, начиная с заданного первого байта S. Преобразованное 16-ричное значение далее пересылается в D, начиная от заданной цифры. Первый байт (старшие или младшие 8 бит) количество байт, подлежащих преобразованию, и цифра в D, принимающая первое 16-ричное значение указаны в DI. Если заданы несколько байт, они будут преобразованы по порядку, начиная с указанной половины S и далее, если необходимо, S+1 и S+2.

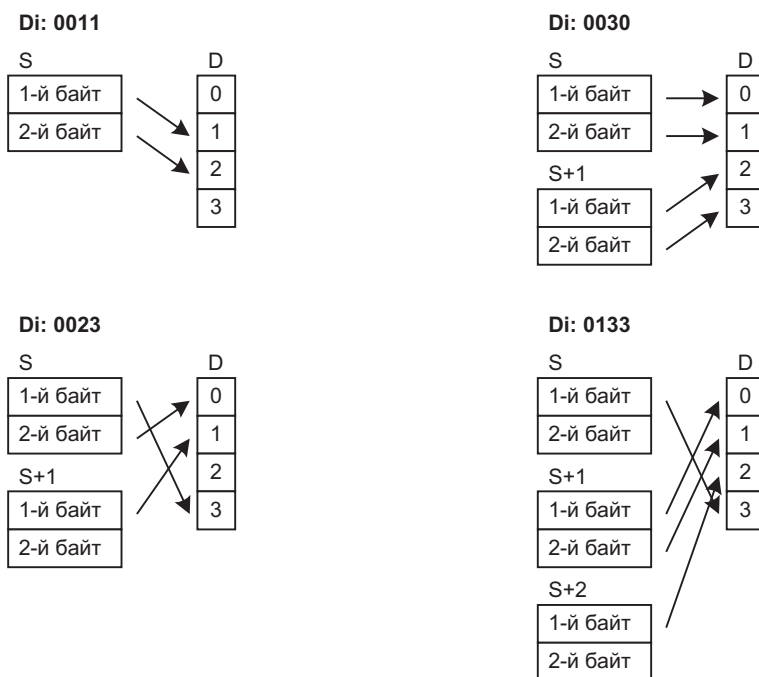
Если задано больше цифр, чем осталось в D (считая от первой заданной), остальные цифры будут браться, начиная с начала D. Цифры в D, которые не принимают преобразованных данных, не изменяются.

**Описатель цифр**

Цифры в DI расположены следующим образом.



На рисунке приведены примеры значения DI и результаты преобразования 8 битовых кодов ASCII в 4-битовые 16-ричные, которые производится при данных значениях.



**Таблица кодов ASCII**

В таблице приведены коды ASCII перед преобразованием и 16-ричные значения после преобразования. Таблицу кодов ASCII см. Приложение G.

Исходные данные									После преобразования					
Код ASCII	Состояние битов (см. прим.)								Цифра	Биты				
30	*	0	1	1	0	0	0	0	0	0	0	0	0	0
31	*	0	1	1	0	0	0	1	1	0	0	0	0	1
32	*	0	1	1	0	0	1	0	2	0	0	1	0	
33	*	0	1	1	0	0	1	1	3	0	0	1	1	
34	*	0	1	1	0	1	0	0	4	0	1	0	0	
35	*	0	1	1	0	1	0	1	5	0	1	0	1	
36	*	0	1	1	0	1	1	0	6	0	1	1	0	
37	*	0	1	1	0	1	1	1	7	0	1	1	1	
38	*	0	1	1	1	0	0	0	8	1	0	0	0	
39	*	0	1	1	1	0	0	1	9	1	0	0	1	
41	*	1	0	1	0	0	0	0	A	1	0	1	0	
42	*	1	0	1	0	0	1	1	B	1	0	1	1	
43	*	1	0	1	0	0	1	0	C	1	1	0	0	
44	*	1	0	1	0	1	0	1	D	1	1	0	1	
45	*	1	0	1	0	1	0	0	E	1	1	1	0	
46	*	1	0	1	0	1	1	1	F	1	1	1	1	

**Замечание** Старший бит каждого кода ASCII устанавливается в соответствии с методом контроля по четности.

**Контроль четности**

Старший бит каждого знака ASCII (2 цифры = 8 бит) может автоматически настраиваться на четность или нечетность.

Если контроль по четности не задан, старший бит всегда = 0.

Когда контроль четности установлен на четно или нечетно, старший бит будет принимать такое значение, чтобы общее количество битов, находящихся в состоянии 1, было четным или нечетным.

Если контроль по четности кода ASCII в S не совпадает с заданным в DI, флаг ошибки (SR 25503) устанавливается = 1 и команда не выполняется.

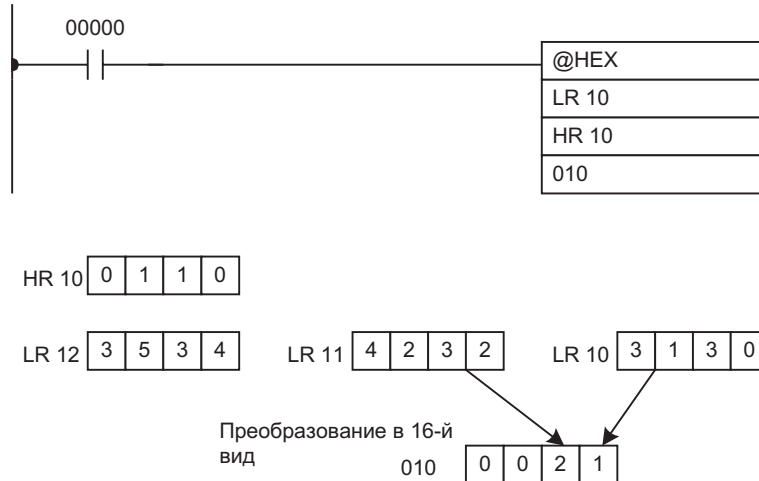
**Флаги**

**ER:** Некорректный определитель цифр или адрес приемника выходит за границы области данных.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

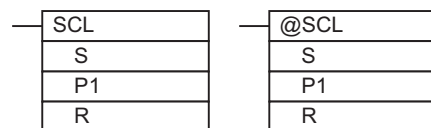
**Пример**

В следующие примере 2-й бит LR 10 и 1-й бит LR 11 преобразуются в 16-ричные значения и это значение записывается в первый и второй байты IR 010.



5.18.12 SCL - Масштабирование

**Обозначение на ПК**



Контроллер	Номер инструкции	Операнды		
		S	P1	R
C200H□-CPU□□-E	-	источник	первое слово параметров	результат
C200H□-CPU□□-ZE	194	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
		EM, #	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

**Ограничения**

P1 и P1+2 должны быть в двоично-десятичном виде.

P1..P1+3 должны находиться в одной области данных.

P1+1 и P1+3 должны иметь разные значения.

**Описание**

SCL служит для линейного преобразования 16-ричного числа из 4 цифр в двоично-десятичное число из 4 цифр. В отличие от команды BCD, которая преобразует 16-ричное число из 4 цифр в двоично-десятичный эквивалент из 4 цифр, SCL может преобразовывать 16-ричное значение с заданным линейным коэффициентом. Линия преобразования задается двумя точками, заданными в слове параметров P1..P1+3.

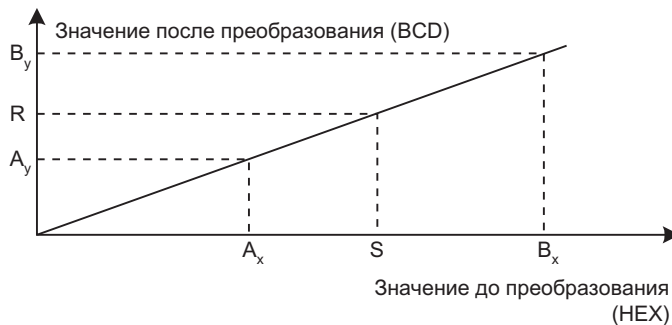
Когда условие исполнения = 0, SCL не выполняется. Когда условие исполнения = 1, SCL преобразует 16-ричное число из 4 цифр источника S в двоично-десятичное значение из 4 цифр с коэффициентом, определяемом на линии, заданной точками (P1, P1+1) и (P1+2, P1+3) и помещает результат в R. Результат округляется до ближайшего числа. Если результат меньше 0, в R заносится 0, если результат больше 9999, в R заносится 9999.



В таблице показаны функции и диапазоны слов параметров.

Параметр	Функция	Диапазон	Комментарии
P1	Точка #1 на оси двоично-десятичной (A <sub>Y</sub> )	0000 - 9999	---
P1+1	Точка #1 на оси 16-ричной (A <sub>X</sub> )	0000 - FFFF	Не задавать P1+1 = P1+3
P1+2	Точка #2 на оси двоично-десятичной (B <sub>Y</sub> )	0000 - 9999	---
P1+3	Точка #2 на оси 16-ричной (B <sub>X</sub> )	0000 - FFFF	Не задавать P1+3 = P1+1

На диаграмме показаны слово источника S, преобразованное в D по линии, заданной точками (A<sub>Y</sub>, A<sub>X</sub>) и (B<sub>Y</sub>, B<sub>X</sub>).



Значения можно посчитать, преобразовав сначала все значения в двоично-десятичный вид, а потом применив формулу:

$$\text{Результат} = B_Y - [(B_Y - A_Y) / (B_X - A_X)] \times (B_X - S)$$

**Флаги**

**ER:** Значение P1+1 равно значению P1+3.

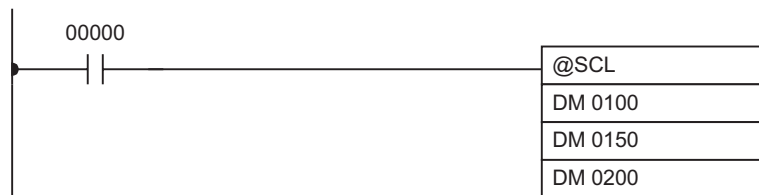
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM)

P1+1 и P1+3 находятся не в одной области данных или другая ошибка задания.

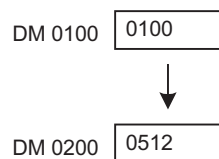
**EQ:** 1, когда результат, R равен 0.

**Пример**

Когда 00000 станет = 0, данные в двоично-десятичном источнике DM 0100 (#0100) преобразуются в 16-ричные согласно параметрам в DM 0150 - DM 0153. Результат (#0512) далее заносится в DM 0200.

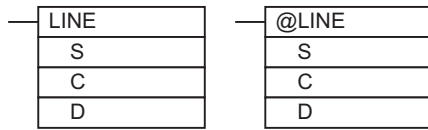


DM 0150	0010
DM 0151	0005
DM 0152	0050
DM 0153	0019



5.18.13 LINE - Преобразование Столбец-в-строку

Обозначение на ПКС



Контроллер	Номер инструкции №	Операнды		
		S первое слово из 16 слов источника	C	D приемник
C200H-CPU-E	63	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM
C200H-CPU-ZE	063	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM

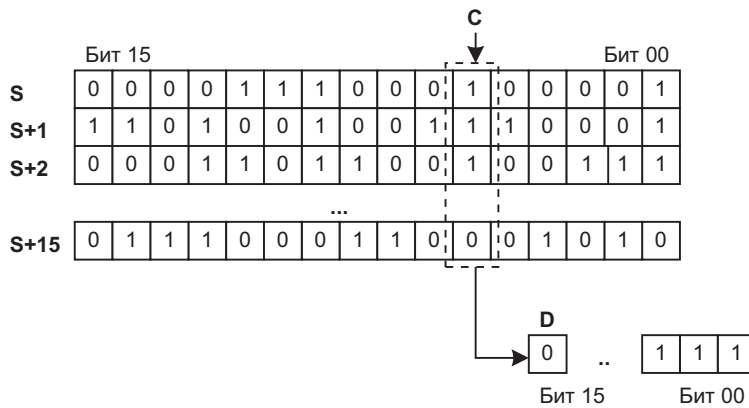
Ограничения

S и S+15 должны находиться в одной области данных.

D должно быть двоично-десятичным числом между #0000 и #0015.

Описание

Когда условие исполнение = 0, LINE не выполняется. Когда условие исполнения = 1, LINE63 копирует столбец битов C из блока 16 слов (от S до S+15) в 16 бит слова D (00..15).



Флаги

**ER:** Описатель битов столбцов C не в двоично-десятичном виде, или определяет несуществующий бит (т.е. имеет значение вне зоны 00 и 15).

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

**EQ:** 1, когда содержание D равно нулю; в противном случае = 0.

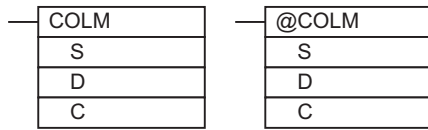
Пример

Следующий пример показывает, как использовать LINE для пересылки битов из столбца 7 из блока (IR 100 - IR 115) в DM 0100.



### 5.18.14 COLM - Преобразование Строка-в-столбец

#### Обозначение на ПКС



Контроллер	Номер инструкции №	Операнды		
		S источник	D	C
C200H□-CPU□□-E	64	IR, SR, HR, AR, LR, TC, DM	IR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM, #
C200H□-CPU□□-ZE	064	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM, #

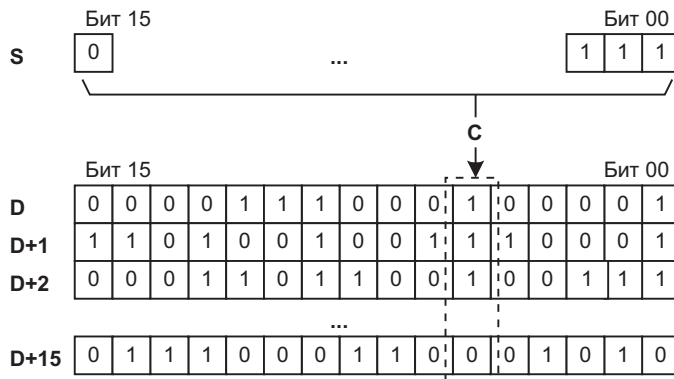
#### Ограничения

D и D+15 должны находиться в одной области данных.

C должно быть двоично-десятичным числом между #0000 и #0015.

#### Описание

Когда условие исполнение = 0, COLM не выполняется. Когда условие исполнения = 1, COLM копирует 16 битов слова S (00..15) в столбец битов, C, блока из 16 слов (D..D+15).



#### Флаги

**ER:** Описатель битов столбцов C не в двоично-десятичном виде, или определяет несуществующий бит (т.е. имеет значение вне зоны 00 и 15).

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

**EQ:** 1, когда содержание S равно нулю; в противном случае = 0.

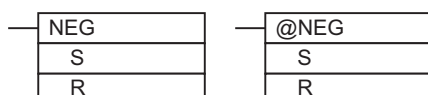
#### Пример

Следующий пример показывает, как использовать COLM для пересылки содержания слов DM 0100 (00 - 15) в столбец битов 15 блока DM 0200 - DM 0215.



### 5.18.15 NEG - Дополнение до 2

#### Обозначение на ПКС



Контроллер	Номер инструкции	Операнды		
		№	S	R
			источник	результат
C200H□-CPU□□-E	-		IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	160		IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Описание**

Преобразует 4-разрядное 16-ричное содержимое источника (S) в дополнение до 2 и выдает результат в слово R. Операция аналогична вычитанию S из 0000 и выдачи результата в R. Высчитывается абсолютное значение в двоичном виде со знаком.

Если содержание S = 0000, в R запишется нуль и флаг EQ (SR 25506) установится в 1.

Если содержание S = 8000, R также будет равен 8000 и флаг UF (SR 25405) установится в 1.

**Замечание** Подробности о 16-битовых двоичных значениях со знаком см. 3.2

**Флаги**

**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

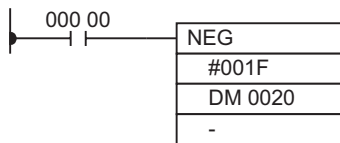
**EQ:** 1, когда содержание R равно нулю; в противном случае = 0.

**UF:** 1, когда содержание S = 8000; в противном случае = 0.

**N:** 1, когда бит 15 в R = 1; в противном случае = 0.

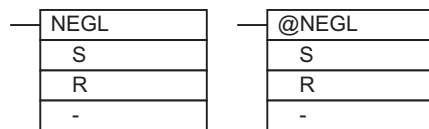
**Пример**

Следующий пример показывает, как использовать NEG для определения дополнения до 2 числа 001F и выдачи результата в DM 0020.



**5.18.16 NEGL - Дополнение до 2 слова двойной длины**

**Обозначение на ПКС**



Контроллер	Номер инструкции	Операнды		
		№	S	R
			первое слово источника	первое слово результата
C200H□-CPU□□-E	-		IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	161		IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

**Ограничения**

S и S+1, должны находиться в одной области данных.

R и R+1, должны находиться в одной области данных.

**Описание**

Преобразует 8-разрядное 16-ричное содержимое слов источника (S и S+1) в дополнение до 2 и выдает результат в слова R и R+1. Операция аналогична вычитанию 8-разрядного

числа S, S+1 из 0000 0000 и выдачи результата в R, R+1. Вычисляется абсолютное значение в двоичном виде со знаком.

Если содержание S = 0000 0000, содержание R также будет 0000 0000 и флаг EQ (SR 25506) установится в 1.

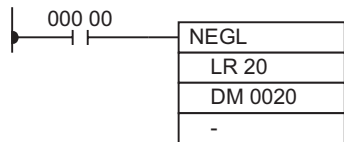
Если содержание S = 8000 0000, содержание R также будет равно 8000 0000 и флаг UF (SR 25405) установится в 1.

**Флаги**

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- EQ:** 1, когда содержание R+1, R равно нулю; в противном случае = 0.
- UF:** 1, когда содержание S+1, S = 8000 0000; в противном случае = 0.
- N:** 1, когда бит 15 в R+1 = 1; в противном случае = 0.

**Пример**

Следующий пример показывает, как использовать NEGL для определения дополнения до 2 содержания слов LR 21, LR 20 (001F FFFF) и выдачи результата в DM 0021, DM 0020.



0000	0000
S+1: LR 24    S: LR 20	
001F	FFFF
R+1: DM 0021    R: DM 0020	
FFE0	0001

## 5.19 Символьные арифметические команды

Символьные арифметические команды осуществляют арифметические операции над числами двоичными, двоично-десятичными и с плавающей запятой.

### 5.19.1 Сложение двоичных чисел: +, +L, +C, +CL

**Сложение знаковых двоичных чисел без учета флага переноса: +**

**Обозначение на РКС**

+	@+
Au	Au
Ad	Ad
R	R

Номер инструкции		Операнды		
Контроллер	Nr	Au	Ad	R
		первое слагаемое	второе слагаемое	результат
C200H-CPU-E				
C200H-CPU-ZE	400	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Сложение знаковых двоичных чисел двойной длины без учета флага переноса: +L**

**Обозначение на РКС**

+L	@+L
Au	Au
Ad	Ad
R	R

Номер инструкции		Операнды		
Контроллер	Nr	Au	Ad	R
		первое слово первого слагаемого	второе слово второго слагаемого	результат
C200H-CPU-E				
C200H-CPU-ZE	401	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

**Сложение знаковых двоичных чисел с учетом флага переноса: +C**

**Обозначение на РКС**

+C	@+C
Au	Au
Ad	Ad
R	R

Номер инструкции		Операнды		
Контроллер	Nr	Au	Ad	R
		первое слагаемое	второе слагаемое	результат
C200H-CPU-E				
C200H-CPU-ZE	402	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Сложение знаковых двоичных чисел двойной длины с учетом флага переноса: +CL**

**Обозначение на РКС**

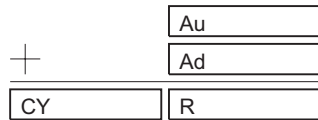
+CL	@+CL
Au	Au
Ad	Ad
R	R

Номер инструкции		Операнды		
Контроллер	Nr	Au	Ad	R
		первое слово первого слагаемого	второе слово второго слагаемого	результат
C200H-CPU-E				
C200H-CPU-ZE	403	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

**Описание**

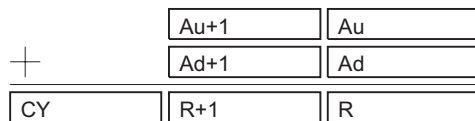
**Сложение знаковых двоичных чисел без учета флага переноса: +**

Когда условие исполнения = 0, + не выполняется. Когда условие исполнения = 1, + складывает содержимое слов AU и AD и помещает результат в R. CY установится в 1, когда результат больше FFFF.



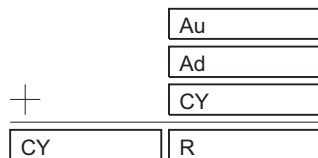
**Сложение знаковых двоичных чисел двойной длины без учета флага переноса: +L**

Когда условие исполнения = 0, +L не выполняется. Когда условие исполнения = 1, +L складывает 8-разрядное число из AU+1, AU и 8-разрядное число из AD+1, AD и помещает результат в R, R+1. CY установится в 1, когда результат больше FFFF FFFF.



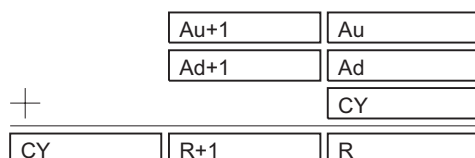
**Сложение знаковых двоичных чисел с учетом флага переноса: +C**

Когда условие исполнения = 0, +C не выполняется. Когда условие исполнения = 1, +C складывает содержимое слов AU, AD и CY и помещает результат в R. CY установится в 1, когда результат больше FFFF.



**Сложение знаковых двоичных чисел двойной длины с учетом флага переноса: +CL**

Когда условие исполнения = 0, +CL не выполняется. Когда условие исполнения = 1, +CL складывает 8-разрядное число из AU+1, AU и 8-разрядное число из AD+1, AD и перенос CY и помещает результат в R, R+1. CY установится в 1, когда результат больше FFFF FFFF.



**Флаги**

- ER:** Содержимое слова, в котором находится адрес для косвенной адресации к области DM/EM, не в двоично-десятичном виде или выходит за границу области DM/EM.
- CY:** Результат больше FFFF или FFFF FFFF.
- EQ:** Результат равен нулю.
- OF** AU( AU+1) и AD(Ad+1) оба положительны, а результат отрицателен.
- UF** AU( AU+1) и AD(Ad+1) оба отрицательны, а результат положителен.
- N** Показывает состояние бита 15 R или R+1

**Применение команд сложения двоичных чисел со знаком**

Диапазон чисел со знаком: в десятичном виде -32 768..32 767 (-2 147 483 648..2 147 483 647 для команд «двойной длины»), в шестнадцатичном виде 8000..FFFF и 0000..7FFF (8000 0000..FFFF FFFF и 0000 0000.... 7FFF FFFF для команд «двойной длины»)

Отрицательные числа выражаются как дополнение до 2. Если результат сложения лежит в диапазоне 8000..FFFF, это означает отрицательное число и флаг отрицательных значений (SR 25402) будет включен в 1.

Когда AU и AD оба положительны, а результат отрицателен, Флаг переполнения + (SR 25404) = 1. Когда AU и AD оба отрицательны, а результат положителен, Флаг переполнения - (SR 25405) = 1.

Если в результате сложения появился перенос, Флаг переноса станет = 1.

Диапазон двоичных чисел без знака: 0000..FFFF (0000 0000..FFFF FFFF для команд «двойной длины»). так что в десятичном виде диапазон: 0..65 535 (0..4 294 967 295).

### 5.19.2 Сложение двоично-десятичных чисел: +B/ +BL/+BC/+BCL

**Сложение двоично-десятичных чисел без учета флага переноса: +B**

**Обозначение на РКС**

+B	@+B
Au	Au
Ad	Ad
R	R

Номер инструкции		Операнды		
Контроллер	№	Au	Ad	R
		первое слагаемое	второе слагаемое	результат
C200H-CPU□□-E				
C200H-CPU□□-ZE	404	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Сложение двоично-десятичных чисел двойной длины без учета флага переноса: +BL**

**Обозначение на РКС**

+BL	@+BL
Au	Au
Ad	Ad
R	R

Номер инструкции		Операнды		
Контроллер	№	Au	Ad	R
		первое слово первого слагаемого	второе слово второго слагаемого	результат
C200H-CPU□□-E				
C200H-CPU□□-ZE	405	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

**Сложение двоично-десятичных чисел с учетом флага переноса: +BC**

**Обозначение на РКС**

+BC	@+BC
Au	Au
Ad	Ad
R	R

Номер инструкции		Операнды		
Контроллер	№	Au	Ad	R
		первое слагаемое	второе слагаемое	результат
C200H-CPU□□-E				
C200H-CPU□□-ZE	406	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Сложение двоично-десятичных чисел двойной длины с учетом флага переноса: +BCL**

**Обозначение на РКС**

+BCL	@+BCL
Au	Au
Ad	Ad
R	R



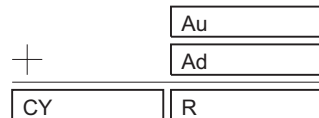
## 5.19 Символьные арифметические команды

Номер инструкции		Операнды		
Контроллер	№	Au	Ad	R
		первое слово первого слагаемого	второе слово второго слагаемого	результат
C200H□-CPU□□-E				
C200H□-CPU□□-ZE	407	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

### Описание

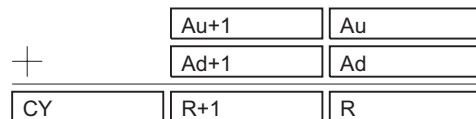
#### Сложение двоично-десятичных чисел без учета флага переноса: +B

Когда условие исполнение = 0, +B не выполняется. Когда условие исполнения = 1, +B складывает содержимое слов AU и AD и помещает результат в R. CY установится в 1, когда результат больше 9999.



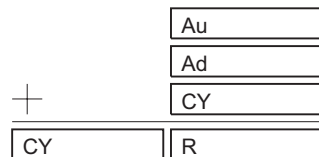
#### Сложение двоично-десятичных чисел двойной длины без учета флага переноса: +BL

Когда условие исполнение = 0, +BL не выполняется. Когда условие исполнения = 1, +BL складывает 8-разрядное число из AU+1, AU и 8-разрядное число из AD+1, AD и помещает результат в R, R+1. CY установится в 1, когда результат больше 9999 9999.



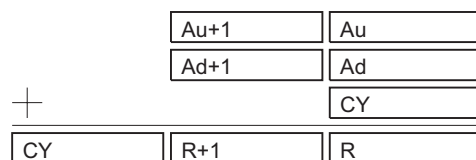
#### Сложение двоично-десятичных чисел с учетом флага переноса: +BC

Когда условие исполнение = 0, +BC не выполняется. Когда условие исполнения = 1, +BC складывает содержимое слов AU, AD и CY и помещает результат в R. CY установится в 1, когда результат больше 9999.



#### Сложение двоично-десятичных чисел двойной длины с учетом флага переноса: +BCL

Когда условие исполнение = 0, +BCL не выполняется. Когда условие исполнения = 1, +BCL складывает 8-разрядное число из AU+1, AU и 8-разрядное число из AD+1, AD и перенос CY и помещает результат в R, R+1. CY установится в 1, когда результат больше 9999 9999.



### Предосторожности

AU и AD (или AU+1, AU и AD+1, AD) должны быть двоично-десятичными числами. Если формат числе другой, Флаг ошибки (SR 25503) станет = 1 и команда не будет выполнена.

### Флаги

- ER:** AU и AD (или AU+1, AU и AD+1, AD) не двоично-десятичные числа. Содержимое слова, в котором находится адрес для косвенной адресации к области DM/EM, не двоично-десятичное число или выходит за границу области DM/EM.

**СУ:** Результат больше 9999 или 9999 9999.

**EQ:** Результат равен нулю.

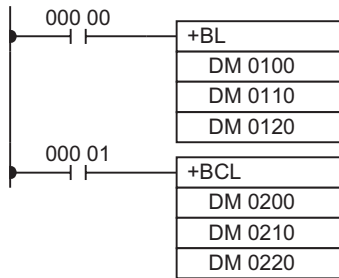
**Пример**

**Операция +BL**

Когда IR 00000 = 1 в следующем примере, содержание DM 0101 и DM 0100 складывается с содержимым DM 0111 и DM 0110 и 8-разрядный двоично-десятичный результат выдается в DM 0121 и DM 0120.

**Операция +BCL**

Когда IR 00001 = 1 в следующем примере, содержание DM 0201 и DM 0200 складывается с содержимым DM 0211 и DM 0210 и 8-разрядный двоично-десятичный результат, включая перенос, выдается в DM 0221 и DM 0220.

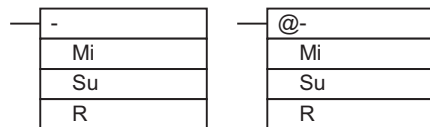


Адрес	Инструкция	Операнд
00000	LD	000 00
00001	+BL	DM 0100
		DM 0110
		DM 0120
00002	LD	000 01
00003	+BCL	DM 0200
		DM 0210
		DM 0220

**5.19.3 Вычитание двоичных чисел: -, -L, -C, -CL**

**Вычитание двоичных чисел без учета флага переноса: -**

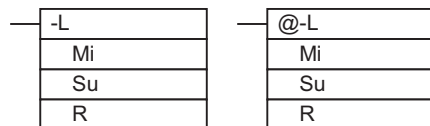
**Обозначение на РКС**



Номер инструкции		Операнды		
Контроллер	Nr	Mi	Su	R
		уменьшаемое	вычитаемое	результат
C200H□-CPU□□-E				
C200H□-CPU□□-ZE	410	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Вычитание двоичных чисел двойной длины без учета флага переноса: -L**

**Обозначение на РКС**



## 5.19 Символьные арифметические команды

Номер инструкции		Операнды		
Контроллер	№	Mi	Su	R
		первое слово уменьшаемого	первое слово вычитаемого	результат
C200H-CPU-E				
C200H-CPU-ZE	411	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

### Вычитание двоичных чисел с учетом флага переноса: -C

#### Обозначение на ПКС

-C	@-C
Mi	Mi
Su	Su
R	R

Номер инструкции		Операнды		
Контроллер	№	Mi	Su	R
		уменьшаемое	вычитаемое	результат
C200H-CPU-E				
C200H-CPU-ZE	412	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

### Вычитание двоичных чисел двойной длины с учетом флага переноса: -CL

#### Обозначение на ПКС

-CL	@-CL
Mi	Mi
Su	Su
R	R

Номер инструкции		Операнды		
Контроллер	№	Mi	Su	R
		первое слово уменьшаемого	первое слово вычитаемого	результат
C200H-CPU-E				
C200H-CPU-ZE	413	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

#### Описание

##### Вычитание двоичных чисел без учета флага переноса: -

Когда условие исполнения = 0, - не выполняется. Когда условие исполнения = 1, - вычитает содержимое Su из Mi и помещает результат в R. Если вычитание привело к отрицательному переносу, установится CY = 1. Для получения истинного результата, когда результат отрицателен, дополнение до 2, которое помещено в R, нужно вычесть из 0000.

	Mi
	Su
CY	R

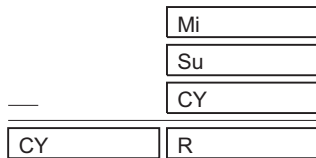
##### Вычитание двоичных чисел двойной длины без учета флага переноса: -L

Когда условие исполнения = 0, -L не выполняется. Когда условие исполнения = 1, -L вычитает 8-разрядное число из Su+1, Su из 8-разрядного числа Mi+1, Mi и помещает результат в R, R+1. Если вычитание привело к отрицательному переносу, установится CY = 1.

	Mi+1	Mi
	Su+1	Su
CY	R+1	R

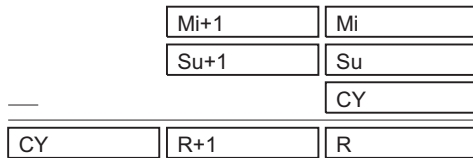
##### Вычитание двоичных чисел с учетом флага переноса: -C

Когда условие исполнения = 0, -C не выполняется. Когда условие исполнения = 1, -C вычитает содержимое Su и CY из Mi и помещает результат в R. Если вычитание привело к отрицательному переносу, установится CY = 1.



**Вычитание двоичных чисел двойной длины с учетом флага переноса: -CL**

Когда условие исполнения = 0, -CL не выполняется. Когда условие исполнения = 1, -CL вычитает 8-разрядное содержимое Su, Su+1 из Mi, Mi+1 и помещает результат в R, R+1. Если вычитание привело к отрицательному переносу, установится CY = 1.



**Флаги**

- ER:** Содержимое слова, в котором находится адрес для косвенной адресации к области DM/EM, не в двоично-десятичном виде, или выходит за границу области DM/EM.
- CY:** Вычитание привело к отрицательному переносу.
- EQ:** Содержание R (или R, R+1 для команд «двойной длины») = 0.
- OF** MI положительно, SU отрицательно, а результат отрицателен.
- UF** MI отрицательно, SU положительно, а результат положителен.
- N** Старший бит слова R (или R+1 для команд «двойной длины») после вычитания = 1.

**Применение команд вычитания двоичных чисел со знаком**

Диапазон чисел со знаком: в десятичном виде -32 768..32 767 (-2 147 483 648..2 147 483 647 для команд «двойной длины»), в шестнадцатиричном виде 8000..FFFF и 0000..7FFF (8000 0000..FFFF FFFF и 0000 0000.... 7FFF FFFF для команд «двойной длины»)

Отрицательные числа выражаются как дополнение до 2. Если результат сложения лежит в диапазоне 8000..FFFF, это означает отрицательное число и флаг отрицательных значений (SR 25402) будет включен в 1.

Когда MI положительно, SU отрицательно, а результат отрицателен, Флаг переполнения + (SR 25404) = 1. Когда MI отрицательно, SU положительно, а результат положителен, Флаг переполнения - (SR 25405) = 1.

Если в результате вычитания появился перенос, Флаг переноса станет = 1.

Диапазон двоичных чисел без знака: 0000..FFFF (0000 0000..FFFF FFFF для команд «двойной длины»), так что в десятичном виде диапазон: 0..65 535 (0..4 294 967 295).

При значениях без знака флаг переноса, равный 1, указывает, что результат отрицателен. Результат выражен дополнением до 2, так что для нахождения истинного результата дополнение до 2 нужно вычесть из 0.

**Пример 1**

	Знаковые данные	Беззнаковые данные
—	FFF	-1
—	0001	+1
	FFFE	-2
Флаг N - ON, флаг CY - OFF	(см. замеч. 1)	65535
		1
		35534
		(см. замеч. 2)

**Пример 2**

	Знаковые данные	Беззнаковые данные
— FFFD	— -3	— 65533
— FFFF	— +1	— 65535
— FFFE	— -1	— -2
Флаг N - ON, флаг CY - ON	(см. замеч. 1)	(см. замеч. 3)

- Замечание**
1. Поскольку Флаг отрицательного результата = 1, результат (FFFE) - отрицательное число (дополнение до 2) и выражено как -2.
  2. Флаг переноса = 0, и результат (FFFE) - положительное число без знака (65 534).
  3. Флаг переноса = 1, так что результат (FFFE) - отрицательное число (дополнение до 2) и после преобразования = -2.

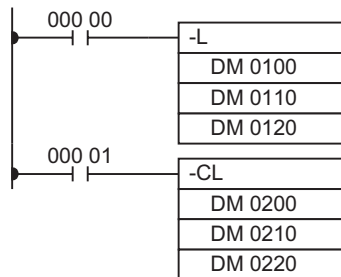
**Пример программирования 1**

**Операция -L**

Когда IR 00000 = 1 в следующем примере, содержание DM 0111 и DM 0110 вычитается из содержимого DM 0101 и DM 0100 и 8-разрядный двоичный результат выдается в DM 0121 DM 0120. CY = 1, если вычитание привело к заему (отрицательному переполнению).

**Операция -CL**

Когда IR 00001 = 1 в следующем примере, содержание DM 0211 и DM 0210 вычитается из содержимого DM 0201 и DM 0200 и 8-разрядный двоичный результат выдается в DM 0221 DM 0220. CY = 1, если вычитание привело к заему (отрицательному переполнению).



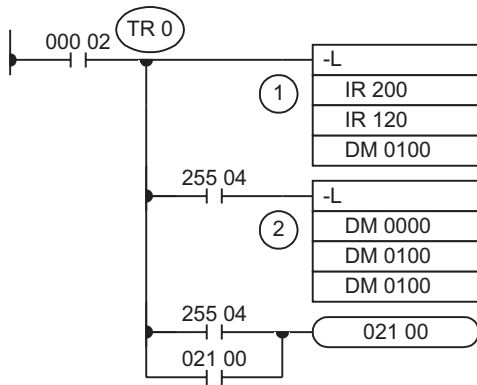
Адрес	Инструкция	Операнд
00000	LD	000 00
00001	-L	DM 0100
		DM 0110
		DM 0120
00002	LD	000 01
00003	-CL	DM 0200
		DM 0210
		DM 0220

**Пример программы 2**

Пример (числа без знака): 20F55A10-B8A360E3= - 97AE06D3.

В данном примере 8-разрядное двоичное число в IR 121, IR 120 вычитается из IR 201, IR 200 и 8-разрядный двоичный результат выдается в DM 0101, DM 0100. Если результат отрицателен, команда, отмеченная (2) будет выполнена и результат будет помещен в DM 0101, DM 0100.

Флаг переноса (SR 25504) будет =1, так что фактическое значение = -97AE06D3. Поскольку содержание DM 0101, DM 0100 отрицательно, CY служит для переключения в 1 бита самоподдержки, который включается в 1 для индикации отрицательного значения.



Адрес	Инструкция	Операнд
00000	LD	000 02
00001	OUT	TR 0
00002	-L	200
		120
		DM 1000
00003	LD	TR 0
00004	AND	SR 255 04
00005	-L	DM 0000
		DM 0100
		DM 0100
00006	LD	TR 0
00007	AND	SR 255 04
00008	OR	021 00
00009	OUT	021 00

**Инструкция 1**

	Mi+1: IR 201	Mi: IR 200
	2 0 F 5	5 A 1 0
	Su+1: IR 121	Su: IR 120
—	B 8 A 3	6 0 E 3
CY	R+1: DM 0101	R: DM 0100
1	6 8 5 1	F 9 2 D

Флаг переноса (SR 25504) =1, так что результат вычитается из 0000 0000 (содержание DM 0001, DM 0000) для получения фактического результата.

**Инструкция 2**

	Mi+1: DM 201	Mi: DM 0000
	0 0 0 0	0 0 0 0
	Su+1: DM 0101	Su: DM 0100
—	6 8 5 1	F 9 2 D
CY	R+1: DM 0101	R: DM 0100
1	9 7 A E	0 6 D 3

**Результат**

	Mi+1: IR 201	Mi: IR 200
	2 0 F 5	5 A 1 0
	Su+1: DM 0101	Su: DM 0100
—	6 8 5 1	F 9 2 D
CY	R+1: DM 0101	R: DM 0100
1	9 7 A E	0 6 D 3

## 5.19.4 Вычитание двоично-десятичных чисел: -B/ -BL/-BC/-BCL

Вычитание двоично-десятичных чисел без учета флага переноса: -B

Обозначение на ПКС

-B	@-B
Mi	Mi
Su	Su
R	R

Номер инструкции		Операнды		
Контроллер	Nr	Mi	Su	R
		уменьшаемое	вычитаемое	результат
C200H□-CPU□□-E				
C200H□-CPU□□-ZE	414	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

Вычитание двоично-десятичных чисел двойной длины без учета флага переноса: -BL

Обозначение на ПКС

-BL	@-BL
Mi	Mi
Su	Su
R	R

Номер инструкции		Операнды		
Контроллер	Nr	Mi	Su	R
		первое слово уменьшаемого	первое слово вычитаемого	результат
C200H□-CPU□□-E				
C200H□-CPU□□-ZE	415	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

Вычитание двоично-десятичных чисел с учетом флага переноса: -BC

Обозначение на ПКС

-BC	@-BC
Mi	Mi
Su	Su
R	R

Номер инструкции		Операнды		
Контроллер	Nr	Mi	Su	R
		уменьшаемое	вычитаемое	результат
C200H□-CPU□□-E				
C200H□-CPU□□-ZE	416	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

Вычитание двоично-десятичных чисел двойной длины с учетом флага переноса: -BCL

Обозначение на ПКС

-BCL	@-BCL
Mi	Mi
Su	Su
R	R

Номер инструкции		Операнды		
Контроллер	Nr	Mi	Su	R
		первое слово уменьшаемого	первое слово вычитаемого	результат
C200H□-CPU□□-E				
C200H□-CPU□□-ZE	417	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

Описание

Вычитание двоично-десятичных чисел без учета флага переноса: -B

Когда условие исполнение = 0, -B не выполняется. Когда условие исполнения = 1, -B вычитает содержимое Su из Mi и помещает результат в R. Если результат

отрицательный, установится  $CY = 1$  и в R будет помещено дополнение до 10 фактического результата. Для преобразования дополнения до 10 в фактический результат вычтите содержимое R из 0000.

	Mi
—	Su
<hr/>	
CY	R

**Вычитание двоично-десятичных чисел двойной длины без учета флага переноса: -BL**

Когда условие исполнение = 0, -BL не выполняется. Когда условие исполнения = 1, -BL вычитает 8-разрядное двоично-десятичное число  $Su+1$ ,  $SU$  из 8-разрядного двоично-десятичного числа  $Mi+1$ ,  $MI$  и помещает результат в R, R+1. Если результат отрицательный, установится  $CY = 1$  и в R будет помещено дополнение до 10 фактического результата. Для преобразования дополнения до 10 в фактический результат вычтите содержимое R из 0000 0000.

	Mi+1	Mi
—	Su+1	Su
<hr/>		
CY	R+1	R

**Вычитание двоично-десятичных чисел с учетом флага переноса: -BC**

Когда условие исполнение = 0, -BC не выполняется. Когда условие исполнения = 1, -BC вычитает содержимое Su и CY из Mi и помещает результат в R. Если результат отрицательный, установится  $CY = 1$  и в R будет помещено дополнение до 10 фактического результата. Для преобразования дополнения до 10 в фактический результат вычтите содержимое R из 0000.

	Mi
	Su
—	CY
<hr/>	
CY	R

**Вычитание двоично-десятичных чисел двойной длины с учетом флага переноса: -BCL**

Когда условие исполнение = 0, -BCL не выполняется. Когда условие исполнения = 1, -BCL вычитает CY и 8-разрядное двоично-десятичное число  $Su+1$ ,  $SU$  из 8-разрядного двоично-десятичного числа  $Mi+1$ ,  $MI$  и помещает результат в R, R+1. Если результат отрицательный, установится  $CY = 1$  и в R будет помещено дополнение до 10 фактического результата. Для преобразования дополнения до 10 в фактический результат вычтите содержимое R из 0000 0000.

	Mi+1	Mi
	Su+1	Su
—		CY
<hr/>		
CY	R+1	R

**Предосторожности**

$Mi$  и  $Su$  (или  $Mi$ ,  $Mi+1$  и  $Su$ ,  $Su+1$ ) должны быть двоично-десятичными числами. Если формат числе другой, Флаг ошибки (SR 25503) станет = 1 и команда не будет выполнена.

**Флаги**

- ER:**  $Mi$  и  $Su$  (или  $Mi$ ,  $Mi+1$  и  $Su$ ,  $Su+1$ ) не двоично-десятичные числа. Содержимое слова, в котором находится адрес для косвенной адресации к области DM/EM, не двоично-десятичное число или выходит за границу области DM/EM.
- CY:** Результат больше 9999 или 9999 9999.



**EQ:** Результат равен нулю.

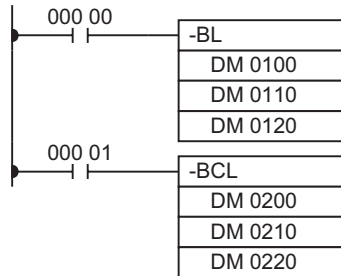
**Пример**

**Операция -BL**

Когда IR 00000 = 1 в следующем примере, содержание DM 0111 и DM 0110 вычитается из содержимого DM 0201 и DM 0200 и 8-разрядный двоично-десятичный результат выдается в DM 0121 DM 0120. Если результат отрицательный, установится флаг CY.

**Операция -BCL**

Когда IR 00001 = 1 в следующем примере, содержание DM 0211 и DM 0210 вычитается из содержимого DM 0201 и DM 0200 и 8-разрядный двоично-десятичный результат, включая перенос, выдается в DM 0221 DM 0220. Если результат отрицательный, установится флаг CY.



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	-BL	DM 0100
		DM 0110
		DM 0120
00002	LD	000 01
00003	-BCL	DM 0200
		DM 0210
		DM 0220

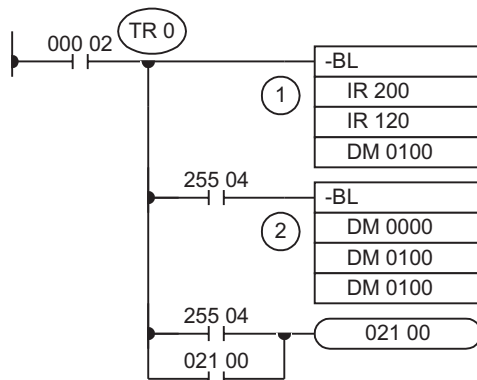
**Пример программы**

Пример: 9 583 960 - 17 072 641 = - 7 488 681.

В данном примере 8-разрядное двоично-десятичное число IR 121, IR 120 вычитается из IR 201, IR 200 и 8-разрядный двоичный результат выдается в DM 0101, DM 0100. Результат отрицателен, поэтому команда, отмеченная (2), будет выполнена и фактический результат будет помещен в DM 0101, DM 0100.

Флаг переноса (SR 25504) будет =1, так что фактическое значение = -7 488 681.

Поскольку содержание DM 0101, DM 0100 отрицательно, CY служит для включения в 1 бита самоподдержки, который включается в 1 для индикации отрицательного значения.



Адрес	Инструкция	Операнд
00000	LD	000 02

## 5.19 Символьные арифметические команды

Адрес	Инструкция	Операнд
00001	OUT	TR 0
00002	-BL	200
		120
		DM 0100
00003	LD	TR 0
00004	AND	SR 255 04
00005	-BL	DM 0000
		DM 0100
		DM 0100
00006	LD	TR 0
00007	AND	SR 255 04
00008	OR	021 00
00009	OUT	021 00

### Инструкция 1

	Mi+1: IR 201	Mi: IR 200
	0 9 5 8	3 9 6 0
	Su+1: IR 211	Su: IR 120
—	1 7 0 7	2 6 4 1
CY	R+1: DM 0101	R: DM 0100
1	9 2 5 1	1 3 1 9

### Инструкция 2

	Mi+1: DM 201	Mi: DM 0000
	0 0 0 0	0 0 0 0
	Su+1: DM 0101	Su: DM 0100
—	9 2 5 1	1 3 1 9
CY	R+1: DM 0101	R: DM 0100
1	0 7 4 8	8 6 8 1

### Результат

	Mi+1: IR 201	Mi: IR 200
	2 0 F 5	5 A 1 0
	Su+1: DM 0101	Su: DM 0100
—	6 8 5 1	F 9 2 D
CY	R+1: DM 0101	R: DM 0100
1	9 7 A E	0 6 D 3

## 5.19.5 Умножение двоичных чисел: \*, \*L, \*U, \*UL

Умножение двоичных чисел со знаком: \*

Обозначение на ПКС

*	@*
Md	Md
Mr	Mr
R	R

Контроллер	Номер инструкции	Операнды		
		Md	Mr	R
C200H-CPU-E		множитель	множитель	результат
C200H-CPU-ZE	420	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Умножение двоичных чисел двойной длины со знаком: \*L**

**Обозначение на ПКС**

*L	@*L
Md	Md
Mr	Mr
R	R

Номер инструкции		Операнды		
Контроллер	Nr	Md	Mr	R
		первое слово первого множителя	первое слово второго множителя	первое слово результата
C200H-CPU-E				
C200H-CPU-ZE	421	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

**Умножение двоичных чисел без знака: \*U**

**Обозначение на ПКС**

*U	@*U
Md	Md
Mr	Mr
R	R

Номер инструкции		Операнды		
Контроллер	Nr	Md	Mr	R
		множитель	множитель	результат
C200H-CPU-E				
C200H-CPU-ZE	422	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Умножение двоичных чисел двойной длины без знака: \*UL**

**Обозначение на ПКС**

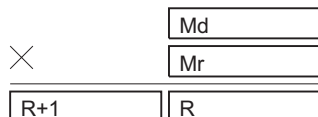
*UL	@*UL
Md	Md
Mr	Mr
R	R

Номер инструкции		Операнды		
Контроллер	Nr	Md	Mr	R
		первое слово первого множителя	первое слово второго множителя	первое слово результата
C200H-CPU-E				
C200H-CPU-ZE	423	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

**Описание**

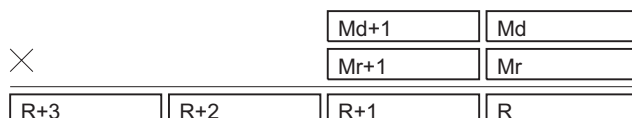
**Умножение двоичных чисел со знаком**

Когда условие исполнения = 0, \* не выполняется. Когда условие исполнения = 1, \* умножает содержимое Md со знаком на содержимое Mr со знаком, помещает правые (младшие) 4 цифры результата в R и левые (старшие) 4 цифры в R+1.



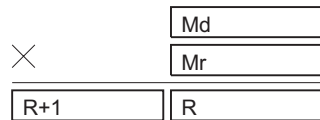
**Умножение двоичных чисел двойной длины со знаком**

Когда условие исполнения = 0, \*L не выполняется. Когда условие исполнения = 1, \*L умножает 8-разрядное число со знаком Md+1, Md на содержимое Mr+1, Mr со знаком и помещает результат в R..R+3.



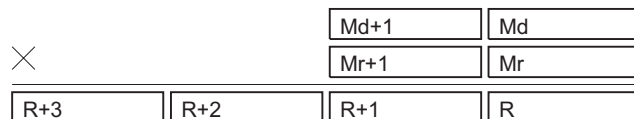
**Умножение двоичных чисел без знака**

Когда условие исполнения = 0, \*U не выполняется. Когда условие исполнения = 1, \*U умножает содержимое Md без знака на содержимое Mr без знака, помещает правые (младшие) 4 цифры результата в R и левые (старшие) 4 цифры в R+1.



**Умножение двоичных чисел двойной длины без знака**

Когда условие исполнения = 0, \*UL не выполняется. Когда условие исполнения = 1, \*UL умножает 8-разрядное число без знака Md+1, Md на содержимое Mr+1, Mr без знака и помещает результат в R..R+3.



**Флаги**

- ER:** Содержимое слова, в котором находится адрес для косвенной адресации к области DM/EM, не в двоично-десятичном виде или выходит за границу области DM/EM.
- EQ:** Результат умножения равен нулю.
- N** Левый (старший) бит слова R+1 (или R+3 для команд «двойной длины») после умножения = 1.

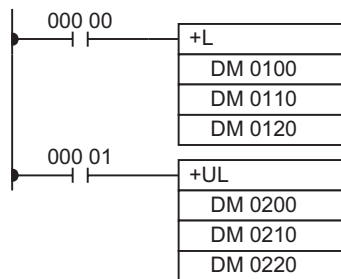
**Пример**

**Операция \*L**

Когда IR 00000 = 1 в следующем примере, содержимое DM 0101 и DM 0100 умножается на содержимое DM 0111 и DM 0110 ( 8-разрядные двоичные числа со знаком) и результат выдается в DM 0123..DM 0120.

**Операция \*UL**

Когда IR 00001 = 1 в следующем примере, содержимое DM 0201 и DM 0200 умножается на содержимое DM 0211 и DM 0210 ( 8-разрядные двоичные числа без знака) и результат выдается в DM 0123..DM 0120.



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	*L	DM 0100
		DM 0110
		DM 0120
00002	LD	000 01
00003	*UL	DM 0200
		DM 0210
		DM 0220

5.19.6 Умножение двоично-десятичных чисел: \*B, \*BL

Умножение двоично-десятичных чисел: \*B

Обозначение на РКС

*B	@*B
Md	Md
Mr	Mr
R	R

Номер инструкции		Операнды		
Контроллер	Nr	Md	Mr	R
		множитель	множитель	результат
C200H-CPU-E				
C200H-CPU-ZE	424	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

Умножение двоично-десятичных чисел двойной длины: \*BL

Обозначение на РКС

*BL	@*BL
Md	Md
Mr	Mr
R	R

Номер инструкции		Операнды		
Контроллер	Nr	Md	Mr	R
		первое слово первого множителя	первое слово второго множителя	первое слово результата
C200H-CPU-E				
C200H-CPU-ZE	425	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

Описание

**Умножение двоично-десятичных чисел**

Когда условие исполнения = 0, \*B не выполняется. Когда условие исполнения = 1, \*B умножает двоично-десятичное число из Md на двоично-десятичное число из Mr и помещает результат в R и R+1.

×	Md
	Mr
R+1	R

**Умножение двоично-десятичных чисел двойной длины**

Когда условие исполнения = 0, \*BL не выполняется. Когда условие исполнения = 1, \*BL умножает 8-разрядное двоично-десятичное число в Md+1, Md на двоично-десятичное число в Mr+1, Mr и помещает результат в R..R+3.

×	Md+1	Md	
	Mr+1	Mr	
R+3	R+2	R+1	R

**Предосторожности**

Md (Md+1) и Mr (Mr+1) должны быть двоично-десятичными числами. Если формат чисел другой, Флаг ошибки (SR 25503) станет = 1 и команда не будет выполнена.

**Флаги**

**ER:** Md (Md+1) и Mr (Mr+1) не двоично-десятичные числа.

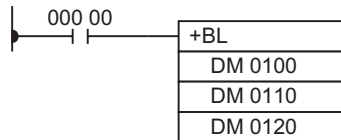
Содержимое слова, в котором находится адрес для косвенной адресации к области DM/EM, не двоично-десятичное число или выходит за границу области DM/EM.

**EQ:** Результат умножения равен нулю.

**Пример**

**Операция \*BL**

Когда IR 00000 = 1 в следующем примере, содержимое DM 0101 и DM 0100 умножается на содержимое DM 0111 и DM 0110 (8-разрядные двоично-десятичные числа) и результат выдается в DM 0123..DM 0120.

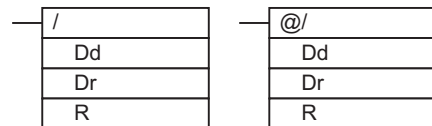


Адрес	Инструкция	Операнд
00000	LD	000 00
00001	*BL	DM 0100
		DM 0110
		DM 0120

**5.19.7 Деление двоичных чисел: /, /L, /U, /UL**

**Деление двоичных чисел со знаком: /**

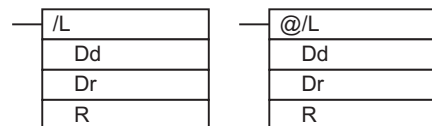
**Обозначение на ПКС**



Номер инструкции		Операнды		
Контроллер	Nr	Dd	Dr	R
		делимое	делитель	первое слово результата
C200H-CPU□□-E				
C200H-CPU□□-ZE	430	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Деление двоичных чисел двойной длины со знаком: /L**

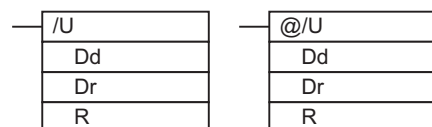
**Обозначение на ПКС**



Номер инструкции		Операнды		
Контроллер	Nr	Dd	Dr	R
		первое слово делимого	первое слово делителя	первое слово результата
C200H-CPU□□-E				
C200H-CPU□□-ZE	431	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

**Деление двоичных чисел без знака: /U**

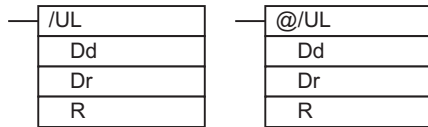
**Обозначение на ПКС**



Номер инструкции		Операнды		
Контроллер	Nr	Dd	Dr	R
		делимое	делитель	первое слово результата
C200H-CPU□□-E				
C200H-CPU□□-ZE	432	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Деление двоичных чисел двойной длины без знака: /UL**

**Обозначение на ПКС**

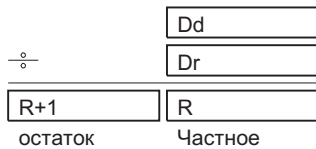


Номер инструкции		Операнды		
Контроллер	№	Dd	Dr	R
		первое слово делимого	первое слово делителя	первое слово результата
C200H□-CPU□□-E				
C200H□-CPU□□-ZE	433	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

**Описание**

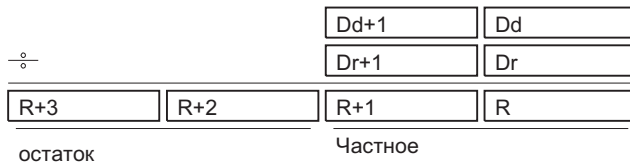
**Деление двоичных чисел со знаком**

Когда условие исполнения = 0, / не выполняется. Когда условие исполнения = 1, / делит двоичное число со знаком в Dd на двоичное число со знаком в Dr и помещает результат в R и R+1: частное в R, остаток в R+1.



**Деление двоичных чисел двойной длины со знаком**

Когда условие исполнения = 0, /L не выполняется. Когда условие исполнения = 1, /L делит 8-разрядное число со знаком Dd+1, Dd на содержимое Dr+1, Dr со знаком и помещает результат в R..R+3: частное в R, R+1 и остаток в R+2, R+3.



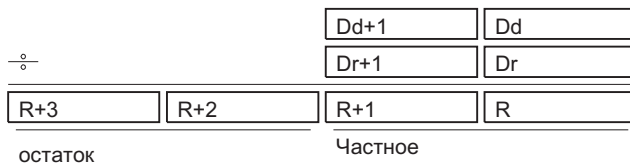
**Деление двоичных чисел без знака**

Когда условие исполнения = 0, /U не выполняется. Когда условие исполнения = 1, / делит двоичное число без знака в Dd на двоичное число без знака в Dr и помещает результат в R и R+1: частное в R, остаток в R+1.



**Деление двоичных чисел двойной длины без знака**

Когда условие исполнения = 0, /UL не выполняется. Когда условие исполнения = 1, /UL делит 8-разрядное число со знаком Dd+1, Dd на содержимое Dr+1, Dr со знаком и помещает результат в R..R+3: частное в R, R+1 и остаток в R+2, R+3.



**Предосторожности**

Dr ( или Dr+1 и Dr) не должны быть равны нулю. Команда не будет выполнена, когда Флаг ошибки (ER) = 1.

**Флаги**

**ER:** D<sub>r</sub> ( или D<sub>r+1</sub> и D<sub>r</sub>) равно нулю.

Содержимое слова, в котором находится адрес для косвенной адресации к области DM/EM, не двоично-десятичное число или выходит за границу области DM/EM.

**EQ:** Результат деления - нуль в частном.

**N:** Левый (старший) бит (MSB) слова R (R+1 для команд «двойной длины» после деления = 1.

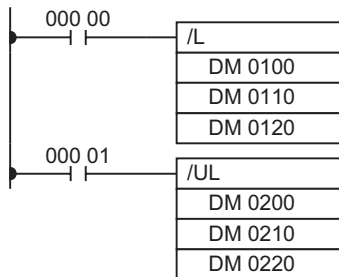
**Пример**

**Операция /L**

Когда IR 00000 = 1 в следующем примере, число со знаком в DM 0101 и DM 0100 делится на число со знаком DM 0111 и DM 0110 ( 8-разрядные двоичные числа). При получении результата частное выдается в DM 0121 и DM 0120, а остаток выдается в DM 0123 и DM 0122.

**Операция /UL**

Когда IR 00001 = 1 в следующем примере, число без знака в DM 0201 и DM 0200 делится на число без знака DM 0211 и DM 0210 ( 8-разрядные двоичные числа). При получении результата частное выдается в DM 0221 и DM 0220, а остаток выдается в DM 0223 и DM 0222.

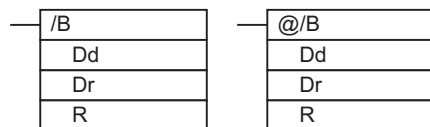


Адрес	Инструкция	Операнд
00000	LD	000 00
00001	/L	DM 0100
		DM 0110
		DM 0120
00002	LD	000 01
00003	/UL	DM 0200
		DM 0210
		DM 0220

**5.19.8 Деление двоично-десятичных чисел: /B, /BL**

**Деление двоично-десятичных: /B**

**Обозначение на ПКС**

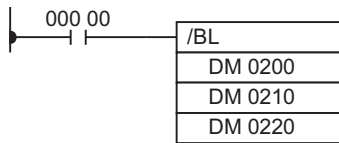


Номер инструкции		Операнды		
Контроллер	Nr	Dd	Dr	R
C200H□-CPU□□-E		делимое	делитель	первое слово результата
C200H□-CPU□□-ZE	432	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM





## 5.19 Символьные арифметические команды



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	/BL	DM 0200
		DM 0210
		DM 0220

## 5.20 Команды вычислений с двоично-десятичными числами

Команды двоично-десятичных вычислений - INC, DEC, ADD, ADDL, SUB, SUBL, MUL, DIV, DIVL, FDIV, ROOT - выполняют арифметические операции над двоично-десятичными числами.

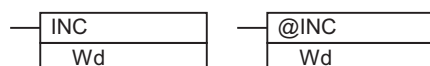
Для INC и DEC слова источника и результата те же самые. Т.е. содержимое слова источника заменяется на результат команды. Все другие команды изменяют только содержание слов, куда помещается результат, т. е. содержимое слов источника одинаковы до и после выполнения команды.

STC и CLC, которые устанавливают и сбрасывают флаг переноса, включены в эту группу, поскольку большинство операций над двоично-десятичными числами используют флаг переноса (CY). Двоичные вычисления и операции сдвига также используют флаг переноса.

Команды сложения и вычитания используют CY и при вычислении и при выдаче результата. Не забудьте очистить CY, если предыдущее значение не используется в вычислении, и используйте результаты, если требуется, до того, как CY могут изменить выполнение какой-либо другой команды.

### 5.20.1 INC - Инкремент

#### Обозначение на ПКК



Номер инструкции		Операнды		
Контроллер	№	Wd		
		инкрементируемое слово		
C200H□-CPU□□-E	38	IR, SR, HR, AR, LR, DM		
C200H□-CPU□□-ZE	038	IR, SR, HR, AR, LR, DM, EM		

#### Описание

Когда условие исполнения = 0, INC не выполняется. Когда условие исполнения = 1, INC инкрементирует (увеличивает на 1) WD, без воздействия на перенос (CY).

#### Флаги

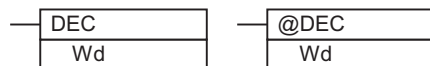
**ER:** WD не в двоично-десятичном виде.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

**EQ:** 1, когда инкрементированный результат равен нулю.

### 5.20.2 DEC - Декремент двоично-десятичного числа

#### Обозначение на ПКК



Номер инструкции		Операнды		
Контроллер	№	Wd		
		декрементируемое слово		
C200H□-CPU□□-E	39	IR, SR, HR, AR, LR, DM		
C200H□-CPU□□-ZE	039	IR, SR, HR, AR, LR, DM, EM		

#### Описание

Когда условие исполнения = 0, DEC не выполняется. Когда условие исполнения = 1, DEC декрементирует (уменьшает на 1) WD, без воздействия на перенос (CY). DEC работает аналогично INC, за исключением того, что декрементирует значение вместо того, чтобы инкрементировать.

#### Флаги

**ER:** WD не в двоично-десятичном виде.

## 5.20 Команды вычислений с двоично-десятичными числами

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

**EQ:** 1, когда декрементированный результат равен нулю.

### 5.20.3 STC - Установка флага переноса

#### Обозначение на ПКС

— 

STC
-----

 — 

@STC
------

	Номер инструкции	Операнды		
Контроллер	№			
C200H□-CPU□□-E	40			
C200H□-CPU□□-ZE	040			

#### Описание

Когда условие исполнение = 0, STC не выполняется. Когда условие исполнения = 1, STC устанавливает в 1 флаг CY (SR 25504).

**Замечание** Таблица с перечислением команд, которые воздействуют на CY, представлена в Приложении С.

### 5.20.4 CLC - Сброс флага переноса

#### Обозначение на ПКС

— 

CLC
-----

 — 

@CLC
------

	Номер инструкции	Операнды		
Контроллер	№			
C200H□-CPU□□-E	41			
C200H□-CPU□□-ZE	041			

#### Описание

Когда условие исполнение = 0, CLC не выполняется. Когда условие исполнения = 1, CLC сбрасывает в 0 флаг CY (SR 25504).

CLC служит для сброса в 0 флаг CY (SR 25504).

CY автоматически сбрасывается в 0 при выполнении команды END в конце каждого цикла.

**Замечание** Таблица с перечислением команд, которые воздействуют на CY, представлена в Приложении С.

### 5.20.5 ADD - Сложение двоично-десятичных чисел

#### Обозначение на ПКС

— 

ADD
Au
Ad
R

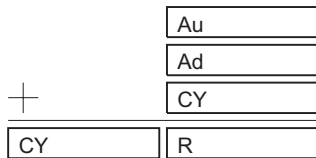
 — 

@ADD
Au
Ad
R

	Номер инструкции	Операнды		
Контроллер	№	Au	Ad	R
		слагаемое	слагаемое	результат
C200H□-CPU□□-E	30	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	030	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

#### Описание

Когда условие исполнение = 0, ADD не выполняется. Когда условие исполнения = 1, ADD складывает содержимое слов AU, AD и CY и выдает результат в R. CY установится в 1, когда результат больше 9999.



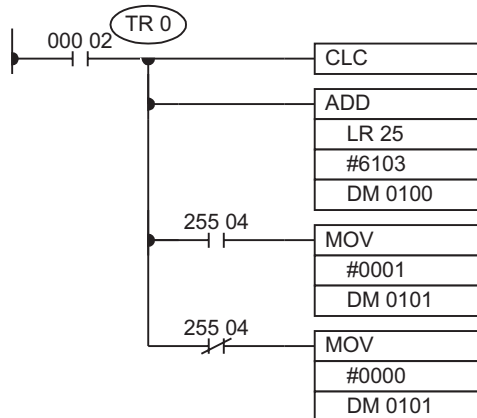
**Флаги**

- ER:** AU и/или AD не двоично-десятичное число.  
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- CY:** 1, когда есть перенос в результате.
- EQ:** 1, когда результат равен нулю.

**Пример**

Если 00002 = 1, программа, представленная на диаграмме, очищает CY командой CLC, складывает содержимое слова IR 030 с константой (6103), помещает результат в DM 0100, затем посылает либо все нули или 0001 в DM 0101 в зависимости от состояния CY (25504). Этим обеспечивается то, что любой перенос с последней цифры сохранится в R+1, чтобы далее можно было рассматривать результат как 8-разрядное число.

Хотя для сложения 8-разрядного числа можно использовать две команды ADD, специально для этой цели предназначена команда ADDL.



Адрес	Инструкция	Операнд
00000	LD	000 02
00001	OUT	TR 0
00002	CLC	
00003	ADD	LR 25
		#6103
		DM 0100
00004	AND	255 04
00005	MOV	#0001
		DM 0101
00006	LD	TR 0
00007	AND NOT	255 04
00008	MOV	#0000
		DM 0101

5.20.6 ADDL - Сложение двоично-десятичных чисел двойной длины

Обозначение на ПКС

ADDL	@ADDL
Au	Au
Ad	Ad
R	R

Контроллер	Номер инструкции	Операнды		
		Au	Ad	R
	Nr	первое слагаемое	второе слагаемое	результат
C200H□-CPU□□-E	54	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	054	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

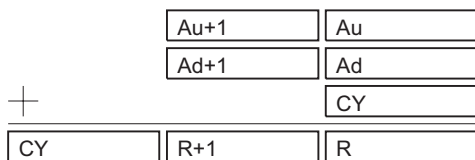
Ограничения

Каждая из следующих пар должны находиться в одной области:

AU и AU+1, AD и AD+1, R и R+1.

Описание

Когда условие исполнение = 0, ADDL не выполняется. Когда условие исполнения = 1, ADDL складывает CY с 8-разрядным числом в AU и AU +1 и с 8-разрядным числом в AD и AD + и выдает результат в R и R+1. CY установится в 1, когда результат больше 9999 9999.



Флаги

**ER:** AU и/или AD не двоично-десятичное число.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

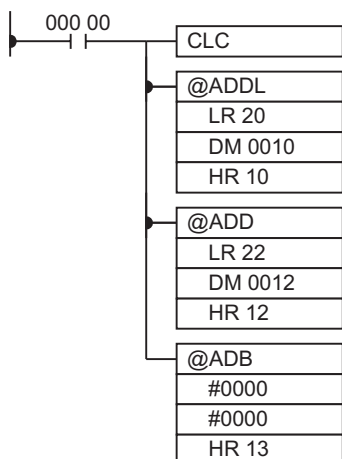
**CY:** 1, когда есть перенос в результате.

**EQ:** 1, когда результат равен нулю.

Пример

Если 00000 = 1, программа, представленная на диаграмме, складывает два 12-разрядных числа, первое из которых лежит в LR 20..LR 22, а второе в DM 0010..DM 0012. Результат заносится в LR 10..LR 13.

Вторая операция сложения (командой ADD) подключает перенос из первого сложения. Перенос от второго сложения помещается в HR 13 командой ADB (см. 5-20-1), к двум константам со всеми нулями для посылки переноса CY от второго сложения в HR 13.



## 5.20 Команды вычислений с двоично-десятичными числами

Адрес	Инструкция	Операнд
00000	LD	000 00
00001	CLC	
00002	@ADDL	LR 20
		DM 0010
		HR 10
00003	@ADD	LR 22
		DM 0012
		HR 12
00004	@ADB	#0000
		#0000
		HR 13

### 5.20.7 SUB - Вычитание двоично-десятичных чисел

#### Обозначение на ПКС

SUB	@SUB
Mi	Mi
Su	Su
R	R

Контроллер	Номер инструкции	Nr	Операнды		
			Mi	Su	R
			уменьшаемое	вычитаемое	результат
C200H□-CPU□□-E	31		IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	031		IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

#### Описание

Когда условие исполнения = 0, SUB не выполняется. Когда условие исполнения = 1, SUB вычитает содержание вычитаемого SU и перенос CY из уменьшаемого MI и выдает результат в R. Если результат отрицательный, CY = 1 и в R пересылается дополнение до 10 фактического значения. Для преобразования дополнения до 10 в истинный результат нужно вычесть R из нуля (см. пример далее).

	Mi
	Su
	CY
CY	R

**Замечание** Команду NEG можно использовать только для преобразования двоичных значений (для двоично-десятичных - нельзя).

#### Флаги

- ER:** MI и/или SU не двоично-десятичное число.  
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- CY:** 1, когда результат отрицательный, т.е. MI меньше, чем SU плюс CY.
- EQ:** 1, когда результат равен нулю.

**Внимание!** Обязательно очистите флаг переноса командой CLC перед исполнением SUB, если предыдущее состояние не требуется, и проверьте состояние CY после вычитания. Если CY = 1 в результате исполнения SUB (т.е., если результат отрицателен), результат выдается как дополнение до 10 фактического значения. Для преобразования результата в фактическое значение нужно вычесть R из нуля.

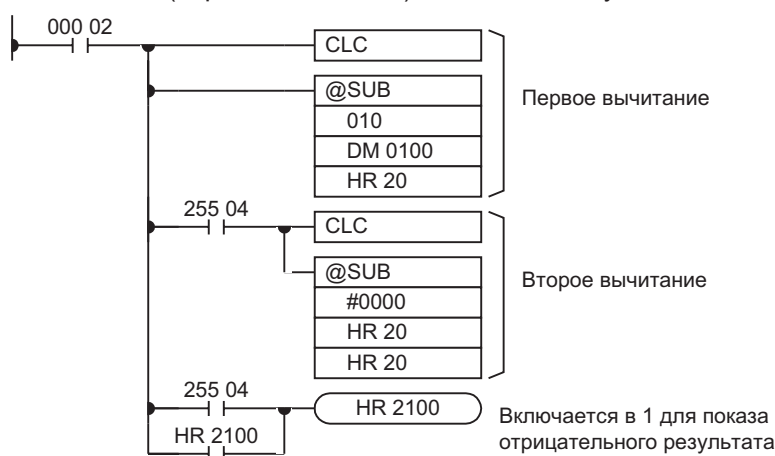
**Пример**

Если 00002 = 1, программа, представленная на диаграмме, очищает CY командой CLC, вычитает содержимое DM 0100 и CY из содержимого 010 и посылает результат в HR 20.

Если CY установлен после исполнения SUB, результат в HR 20 вычитается от нуля (обратите внимание, что снова требуется CLC для получения истинного результата), результат помещается снова в HR 20, а HR 2100 установлен в 1 для указания того, что результат отрицателен.

Если после исполнения SUB CY не установлен, то результат положителен, второе вычитание не требуется, и HR 2100 не устанавливается в 1. HR 2100 запрограммирован как бит самоподдержки, так что изменение состояния CY не установит его в 0 при повторном сканировании программы.

В данном примере применяется вариант SUB фронта 0/1, так что операция вычитания выполняется только раз, каждый раз, когда 00002 устанавливается в 1. Когда требуется новая операция вычитания, требуется хотя бы на 1 цикл установить бит 00002 в состояние 0 (сбросить HR 2100) и затем снова установить в 1.



Адрес	Инструкция	Операнд
00000	LD	000 02
00001	OUT	TR 0
00002	CLC	
00003	@SUB	010
		DM 0100
		HR 20
00004	AND	255 04
00005	CLC	
00006	@SUB	#0000
		HR 20
		HR 20
00007	LD	TR 0
00008	AND	255 04
00009	OR	HR 2100
00010	OUT	HR 2100

Первое и второе вычитание для этой диаграммы показаны ниже, они используют данные из примера для 010 и DM 0100.

**Замечание** Текущая операция SUB включает вычитание SU и CY из 10 000 плюс MI. Для положительного результата старшая цифра опускается. Для отрицательного результата получается дополнение до 10. Процедура получения правильного ответа показана далее.



Первое вычитание	Второе вычитание																
Mi: IR 010 <table border="1" style="margin: 0 auto; text-align: center;"> <tr><td>1</td><td>0</td><td>2</td><td>9</td></tr> </table>	1	0	2	9	Mi: # <table border="1" style="margin: 0 auto; text-align: center;"> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	0	0								
1	0	2	9														
0	0	0	0														
Su: DM 0100 <table border="1" style="margin: 0 auto; text-align: center;"> <tr><td>3</td><td>4</td><td>5</td><td>2</td></tr> </table>	3	4	5	2	Su: HR 20 <table border="1" style="margin: 0 auto; text-align: center;"> <tr><td>7</td><td>5</td><td>7</td><td>7</td></tr> </table>	7	5	7	7								
3	4	5	2														
7	5	7	7														
CY <table border="1" style="margin: 0 auto; text-align: center;"> <tr><td style="width: 40px;"> </td><td style="width: 40px;"> </td><td style="width: 40px;"> </td><td style="width: 40px;"> </td></tr> <tr><td style="text-align: right;">0</td></tr> </table>					0	CY <table border="1" style="margin: 0 auto; text-align: center;"> <tr><td style="width: 40px;"> </td><td style="width: 40px;"> </td><td style="width: 40px;"> </td><td style="width: 40px;"> </td></tr> <tr><td style="text-align: right;">0</td></tr> </table>					0						
0																	
0																	
CY      R: HR 20 <table border="1" style="margin: 0 auto; text-align: center;"> <tr><td style="width: 40px;"> </td><td style="width: 40px;"> </td><td style="width: 40px;"> </td><td style="width: 40px;"> </td></tr> <tr><td style="text-align: right;">1</td><td style="text-align: right;">7</td><td style="text-align: right;">5</td><td style="text-align: right;">7</td></tr> </table>					1	7	5	7	CY      R: HR 20 <table border="1" style="margin: 0 auto; text-align: center;"> <tr><td style="width: 40px;"> </td><td style="width: 40px;"> </td><td style="width: 40px;"> </td><td style="width: 40px;"> </td></tr> <tr><td style="text-align: right;">1</td><td style="text-align: right;">2</td><td style="text-align: right;">4</td><td style="text-align: right;">2</td></tr> </table>					1	2	4	2
1	7	5	7														
1	2	4	2														

В данном случае программа установит HR 2100 в 1 для указания того, что значение в HR 20 отрицательно.

### 5.20.8 SUBL - Вычитание двоично-десятичных чисел двойной длины

#### Обозначение на ПКС

SUBL	@SUBL
Mi	Mi
Su	Su
R	R

Контроллер	Номер инструкции	Операнды			
		Nr	Mi	Su	R
			первое слово уменьшаемого	первое слово вычитаемого	результат
C200H□-CPU□□-E	55		IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	055		IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

#### Ограничения

Каждая из следующих пар должны находиться в одной области:

MI и MI+1, SU и SU+1, R и R+1.

#### Описание

Когда условие исполнение = 0, SUBL не выполняется. Когда условие исполнения = 1, SUBL вычитает CY и 8-разрядное содержимое вычитаемого SU, SU+1 из 8-разрядного уменьшаемого MI, MI+1 и выдает результат в R, R+1. Если результат отрицательный, CY = 1 и в R пересылается дополнение до 10 фактического значения. Для преобразования дополнения до 10 в истинный результат нужно вычесть R из нуля. Поскольку нельзя непосредственно ввести 8-разрядную константу, для создания 8-разрядной константы используйте команду BSET (см. 5.16.3).

Mi+1	Mi
Su+1	Su
	CY
CY	R+1
	R

**Замечание** Команду NEGL можно использовать только для преобразования двоичных значений (для двоично-десятичных - нельзя).

#### Флаги

**ER:** MI, MI+1 и/или SU. SU+1 не двоично-десятичные числа.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

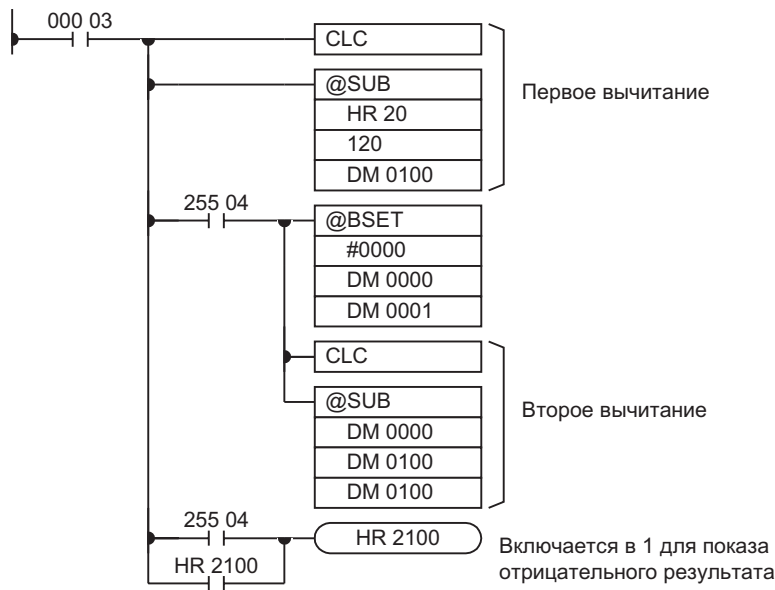
**CY:** 1, когда результат отрицательный, т.е. MI меньше, чем SU.

**EQ:** 1, когда результат равен нулю.

#### Пример

Программа следующего примера работает во многом аналогично вычитанию одного слова. Однако в данном примере для очистки содержания DM 0000и DM 0001 требуется команда BSET, чтобы получить возможность вычитать из нуля отрицательный результат (ввод 8-разрядной константы невозможен).

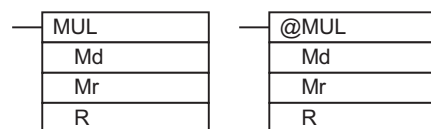
## 5.20 Команды вычислений с двоично-десятичными числами



Адрес	Инструкция	Операнд
00000	LD	000 03
00001	OUT	TR 0
00002	CLC	
00003	@SUBL	HR 20
		120
		DM 0100
00004	AND	255 04
00005	@BSET	#0000
		DM 0000
		DM 0001
00006	CLC	
00007	@SUBL	DM 0000
		DM 0100
		DM 0100
00008	LD	TR 0
00009	AND	255 04
00010	OR	HR 2100
00011	OUT	HR 2100

### 5.20.9 MUL - Умножение двоично-десятичных чисел

#### Обозначение на ПКС



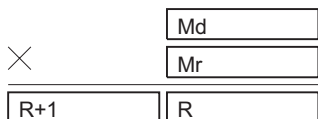
Контроллер	Номер инструкции Nr	Операнды		
		Md множимое	Mr множитель	R первое слово результата
C200H□-CPU□□-E	32	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	032	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

#### Ограничения

R и R+1 должны лежать в одной области данных.

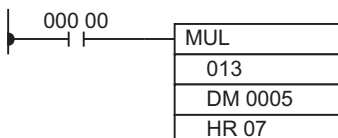
**Описание**

Когда условие исполнения = 0, MUL не выполняется. Когда условие исполнения = 1, MUL умножает MR на MD и посылает результат в R и R+1.

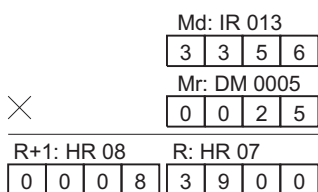


**Пример**

Если при работе следующей диаграммы IR 00000 = 1, содержимое IR 013 и DM 0005 перемножаются и результат посылается в HR 07 и HR 08. Данные и вычисления представлены ниже, после диаграммы.



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	MUL	013
		DM 0005
		HR 07

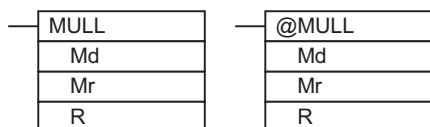


**Флаги**

- ER:** MD и/или MR не двоично-десятичные числа.  
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- EQ:** 1, когда результат равен нулю.

**5.20.10 MULL - Умножение двоично-десятичных чисел двойной длины**

**Обозначение на ПКС**



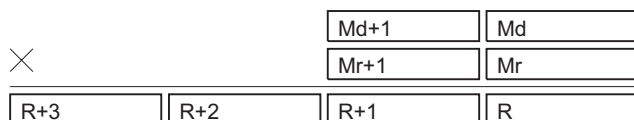
Контроллер	Номер инструкции	Операнды			
		Nr	Md	Mr	R
C200H□-CPU□□-E	56	IR, SR, HR, AR, LR, TC, DM	первое слово множимого	первое слово множителя	первое слово результата
C200H□-CPU□□-ZE	056	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

**Ограничения**

Каждая из следующих пар должны находиться в одной области:  
MD и MD+1, MR и MR+1, а также R..R+3.

**Описание**

Когда условие исполнения = 0, MULL не выполняется. Когда условие исполнения = 1, MULL умножает 8-разрядное содержимое MD, MD+1 на MR, MR+1 и посылает результат в R.. R+3.

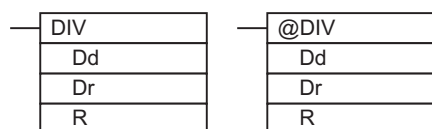


**Флаги**

- ER:** MD, MD+1 и/или MR, MR+1 не двоично-десятичное число.  
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- EQ:** 1, когда результат равен нулю.

**5.20.11 DIV - Деление двоично-десятичных чисел**

**Обозначение на ПКС**



Контроллер	Номер инструкции Nr	Операнды		
		Dd делимое	Dr делитель	R первое слово результата
C200H□-CPU□□-E	33	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	033	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Ограничения**

R и R+1 должны лежать в одной области данных.

**Описание**

Когда условие исполнения = 0, DIV не выполняется. Когда условие исполнения = 1, DIV делит DD на DR и посылает результат в R и R+1: частное в R и остаток в R+1.

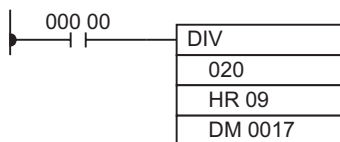


**Флаги**

- ER:** DD и/или DR не двоично-десятичное число или когда DR = #0000.  
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- EQ:** 1, когда результат равен нулю.

**Пример**

Если при работе следующей диаграммы IR 00000 = 1, содержимое IR 020 делится на содержимое HR 09 и результат посылается в DM 0017 и DM 0018. Числовые данные примера и вычисления представлены ниже, после диаграммы.



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	DIV	020
		HR 09
		DM 0017

				Dd: IR 020			
				3	4	5	2
				Dr: Hr 09			
				0	0	0	3
÷							
				R+1: DM 0018    R: DM 0017			
0	0	0	3	1	1	5	2

### 5.20.12 DIVL - Деление двоично-десятичных чисел двойной длины

#### Обозначение на ПКС

DIVL	@DIVL
Dd	Dd
Dr	Dr
R	R

Контроллер	Номер инструкции Nr	Операнды		
		Dd первое слово делимого	Dr первое слово делителя	R результат
C200H□-CPU□□-E	57	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	057	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

#### Ограничения

DD и DD+1 должны лежать в одной области данных, так же, как и DR и DR+1.

R..R+3 должны лежать в одной области данных.

#### Описание

Когда условие исполнения = 0, DIVL не выполняется. Когда условие исполнения = 1, DIVL делит 8-разрядное содержимое DD, DD+1 на содержимое DR, DR+1 и посылает результат в R - R+3: частное в R, R+1 и остаток в R+2, R+3.

				Dd+1	Dd
				Dr+1	Dr
÷					
R+3	R+2	R+1	R		
остаток		Частное			

#### Флаги

**ER:** DR, DR+1 = 0.

DD, DD+1 и/или DR, DR+1 не двоично-десятичное число.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

**EQ:** 1, когда результат равен нулю.

### 5.20.13 FDIV - Деление чисел с плавающей запятой

#### Обозначение на ПКС

FDIV	@FDIV
Dd	Dd
Dr	Dr
R	R

Контроллер	Номер инструкции Nr	Операнды		
		Dd первое слово делимого	Dr первое слово делителя	R результат
C200H□-CPU□□-E	79	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	079	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

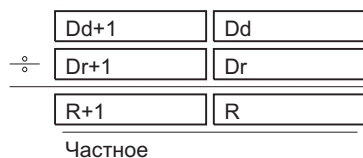
#### Ограничения

Число DR и DR+1 не должно быть равным нулю.

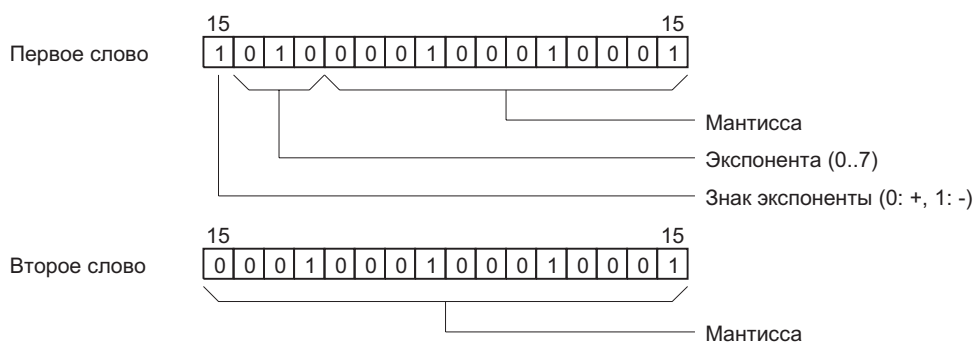
DD и DD+1 должны лежать в одной области данных, так же, как и DR и DR+1, R и R+1.  
 Делимое и делитель должны быть в диапазоне  $0.0000001 \times 10^{-7} \dots 0.9999999 \times 10^7$ .  
 Результат должен быть в диапазоне  $0.1 \times 10^{-7} \dots 0.9999999 \times 10^7$ .

**Описание**

Когда условие исполнения = 0, FDIV не выполняется. Когда условие исполнения = 1, FDIV делит число с плавающей точкой в DD, DD+1 на содержимое DR, DR+1 и посылает результат в R и R+1.



Для представления значений с плавающей точкой 7 младших цифр используются в качестве мантиссы, а старшая цифра служит показателем степени, как показано ниже. Мантисса выражена величиной, меньшей за 1, т.е. до 7 десятичных мест.



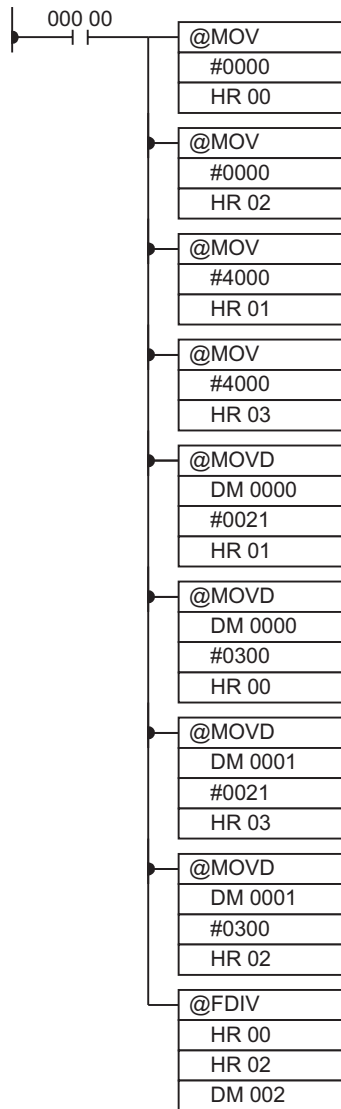
**Флаги**

- ER:** Число в DR и DR+1 = 0.  
 DD, DD+1 и/или DR, DR+1 не двоично-десятичное число.  
 Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- EQ:** 1, когда результат равен нулю.

**Пример**

В следующем примере показано, как разделить два целых 4-разрядных числа (т.е. числа без дробных частей), чтобы получить значение с плавающей точкой.  
 Сначала исходные числа должны быть преобразованы в форму с плавающей точкой. Поскольку исходные числа - без плавающей точки, экспонента будет 4 (т.е.  $3452 = 0.3452 \times 10^4$ ). Все эти шаги предназначены для помещения нужных чисел в последовательные слова для финального деления, включая экспоненты и нули. Перемещения данных для DR и DR+1 такие же. Исходные значения для деления в DM и DM 001. Также показано финальное деление.

## 5.20 Команды вычислений с двоично-десятичными числами



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	@MOV	#0000
		HR 00
00002	@MOV	#0000
		HR 02
00003	@MOV	#4000
		HR 01
00004	@MOV	#4000
		HR 03
00005	@MOVD	DM 0000
		#0021
		HR 01
00006	@MOVD	DM 0000
		#0300
		HR 00
00007	@MOVD	DM 0001
		#0021
		HR 03

## 5.20 Команды вычислений с двоично-десятичными числами

Адрес	Инструкция	Операнд
00008	@MOVD	DM 0001
		#0300
		HR 02
00009	@FDIV	HR 00
		HR 02
		DM 0002

### 5.20.14 ROOT - Квадратный корень

#### Обозначение на ПКС

ROOT	@ROOT
Sq	Sq
R	R

	Номер инструкции	Операнды	
Контроллер	№	Sq	R
		первое слово операнда	результат
C200H-CPU-E	72	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H-CPU-ZE	072	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

#### Ограничения

SQ и SQ+1 должны находиться в одной области данных.

#### Описание

Когда условие исполнения = 0, ROOT не выполняется. Когда условие исполнения = 1, ROOT вычисляет квадратный корень 8-разрядного содержимого SQ, SQ+1 и выдает результат в R. Дробная часть опускается.

#### Флаги

**ER:** SQ или SQ+1 не двоично-десятичное число.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

**EQ:** 1, когда результат равен нулю.

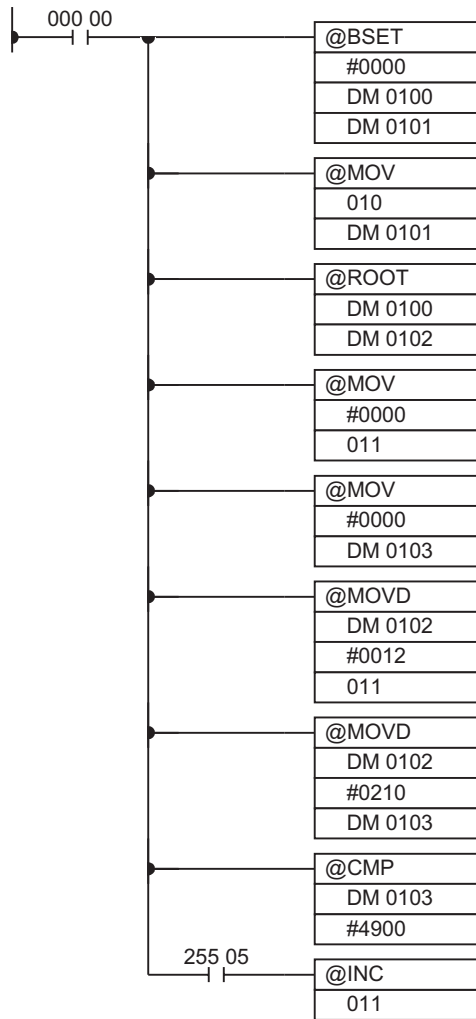
#### Пример

Следующий пример показывает, как взять квадратный корень четырехразрядного числа и затем округлить результат.

Вначале используемые слова очищаются в нули, и затем число, квадратный корень которого вычисляется, помещается в SQ +1. Результат, в котором в два раза больше цифр, чем требуется для ответа (поскольку количество цифр исходного числа было удвоено), помещается в DM 0102, и цифры разделяются на 2 слова, старшие (левые) 2 цифры в IR 011 для ответа и правые (младшие) две цифры в DM 0103, так что ответ в IR 011 можно было при необходимости округлить. Последний шаг - это сравнение значений в DM 0103, чтобы IR 011 можно было увеличить с использованием флага GT (больше чем).



## 5.20 Команды вычислений с двоично-десятичными числами



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	@BSET	#0000
		DM 0100
		DM 0101
00002	@MOV	010
		DM 0101
00003	@ROOT	DM 0100
		DM 0102
00004	@MOV	#0000
		011
00005	@MOV	#0000
		DM 0103
00006	@MOVD	DM 0102
		#0012
		011
00007	@MOVD	DM 0102
		#0210
		DM 0103
00008	@CMP	DM 0103
		#4900

### 5.20 Команды вычислений с двоично-десятичными числами

---

Адрес	Инструкция	Операнд
00009	LD	255 25
00010	@INC	011

## 5.21 Команды вычислений с двоичными числами

Команды вычислений над двоично-десятичными числами - ADB, SBB, MLB, DVB, ADBL, SBBL, MSB, MBSL, DBS, DBSL - все выполняют арифметические операции над 16-ричными числами.

Четыре из этих команд (ADB, SBB, ADBL, SBBL) могут производить действия над числами со знаками и без знака, две (MLB, DVB) действуют только над числами без знака, и четыре (MSB, MBSL, DBS, DBSL) действуют только над числами со знаком.

Команды сложения и вычитания используют CY и при вычислении и при выдаче результата. Не забудьте очистить CY, если предыдущее значение не используется в вычислении, и используйте результаты, если требуется, до того, как CY могут изменить выполнение какой-либо другой команды. Для управления CY можно воспользоваться командами STC и CLC. Подробности см. 5.19.

Команды двоичного сложения и вычитания со знаком используют флаги переполнения + и переполнения - (UF и OF) для того, чтобы определить, не превышен ли допустимый диапазон для 16-битовых или 32-битовых чисел со знаком. Подробности см. 3.2.

### 5.21.1 ADB - Сложение двоичных чисел

#### Обозначение на ПКК

ADB	@ADB
Au	Au
Ad	Ad
R	R

Контроллер	Номер инструкции Nr	Операнды		
		Au	Ad	R
		первое слагаемое	второе слагаемое	результат
C200H□-CPU□□-E	50	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	050	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

#### Описание

Когда условие исполнения = 0, ADB не выполняется. Когда условие исполнения = 1, ADB складывает содержимое слов AU, AD и CY и выдает результат в R. CY установится в 1, когда результат больше FFFF.

	Au
	Ad
+	CY
CY	R

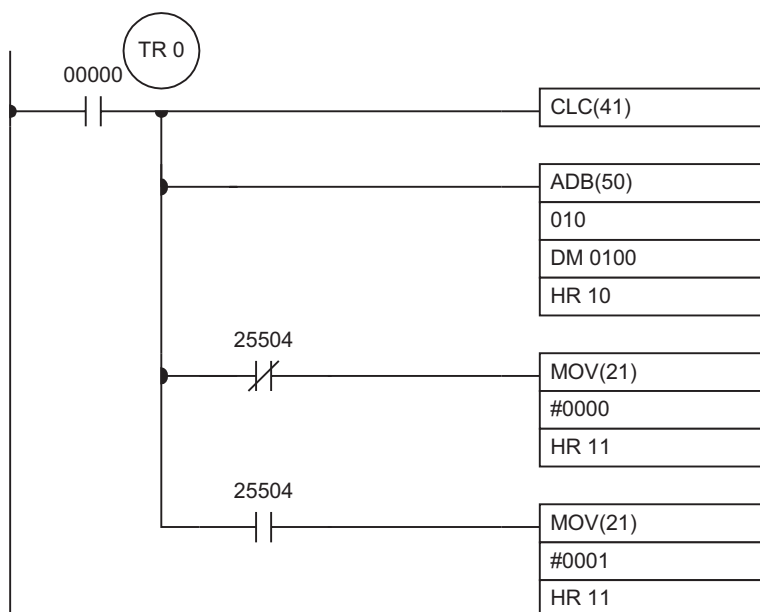
ADB может также складывать двоичные числа со знаком. Флаги переполнение + и переполнение - (SR 25404 и SR 25405) указывают, что результат перешел за верхнюю или нижнюю границу 16-разрядного двоичного числа со знаком. Подробности см. 3.2.

#### Флаги

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- CY:** 1, когда результат больше FFFF.
- EQ:** 1, когда результат равен нулю.
- OF:** 1, когда результат больше + 32 767 (7FFF) (только у CQM1-CPU4?-E)
- UF:** 1, когда результат меньше - 32 768 (8000) (только у CQM1-CPU4?-E)
- N:** 1, когда бит 15 результата = 1.

#### Пример 1: Сложение чисел без знака

Данный пример показывает сложение 4 цифр, причем CY используется для того, чтобы переслать в R+1 либо #0000, либо #0001, чтобы обеспечить сохранение переноса.



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	OUT	TR 0
00002	CLC	
00003	ADB	010
		DM 0100
		HR 10
00004	AND NOT	255 04
00005	MOV	#0000
		HR 11
00006	LD	TR 0
00007	AND	255 04
00008	MOV	#0001
		HR 11

В данном примере  $A6E2 + 80C5 = 127A7$ . Результат - 5-разрядное число, так что CY (SR 25504) = 1 и содержание R+1 устанавливается в #0001.

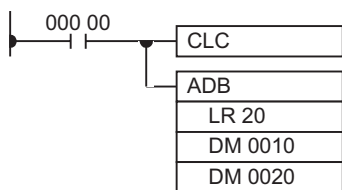
+	Au: IR 010
	A 6 E 2
	Ad: DM 0100
	8 0 C 5
	R+1: HR 11    R: HR 10
	0 0 0 1    2 7 A 7

**Замечание** При сложении могут устанавливаться также флаги UF и OF, но их можно игнорировать, поскольку они нужны только при сложении двоичных данных со знаком.

**Пример 2: Сложение чисел со знаком**

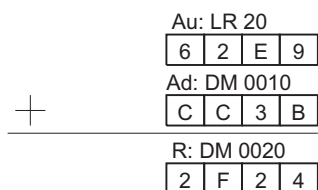
В следующем примере ADB служит для сложения двух 16-битовых двоичных чисел со знаком. (Дополнение до 2 служит для выражения отрицательных значений).

Диапазон 16-битовых двоичных чисел со знаком: -32 768 (8000)..+32 768 (7FFF). Флаг переполнения + (OF: SR 25404) устанавливается в 1, когда результат превышает +32 768 (7FFF), флаг переполнения - (UF: SR 25405) устанавливается в 1, когда результат меньше -32 768 (8000).



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	CLC	
00002	ADB	LR 20
		DM 0010
		DM 0020

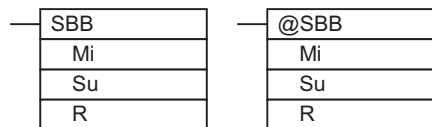
В данном примере  $25\ 321 + (-13\ 253) = 12\ 068$  ( $62E9 + CC3B = 2F24$ ). Ни OF, ни UF не включаются в 1.



**Замечание** Состояние флага CY можно игнорировать, поскольку оно нужно только при сложении 16-ричных чисел без знака.

### 5.21.2 SBB - Вычитание двоичных чисел

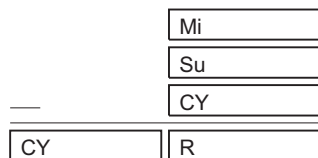
Обозначение на ПКС



Контроллер	Номер инструкции	Операнды		
		Mi	Su	R
C200H□-CPU□□-E	51	уменьшаемое	вычитаемое	результат
C200H□-CPU□□-ZE	051	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM, EM

Описание

Когда условие исполнение = 0, SBB не выполняется. Когда условие исполнения = 1, SBB вычитает содержание вычитаемого SU и перенос CY из уменьшаемого MI и выдает результат в R. Если результат отрицательный, CY = 1 и в R пересылается дополнение до 2 фактического значения.



SBB можно также использовать для вычитания двоичных чисел со знаком. Флаги переполнение + и переполнение - (SR 25404 и SR 25405) указывают, когда результат перешел за верхнюю или нижнюю границу 16-разрядного двоичного числа со знаком. Подробности о Двоичных чисел со знаком см. 3.2.

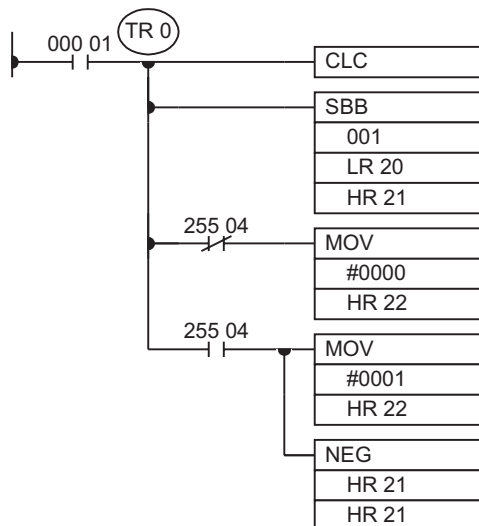
Флаги

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- CY:** 1, когда результат отрицательный, т.е. MI меньше, чем SU плюс CY.

- EQ:** 1, когда результат равен нулю.
- OF:** 1, когда результат больше + 32 767 (7FFF).
- UF:** 1, когда результат меньше - 32 768 (8000).
- N:** 1, когда бит 15 результата установлен в 1.

**Пример 1: Числа без знака**

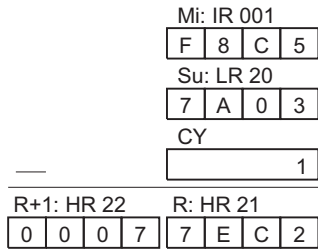
Данный пример показывает вычитание 4-разрядных чисел, CY служит для занесения либо #0000, либо #0001 в R для сохранения переноса.



Адрес	Инструкция	Операнд
00000	LD	000 01
00001	OUT	TR 1
00002	CLC	
00003	SBB	001
		LR 20
		HR 21
00004	AND NOT	255 04
00005	MOV	#0000
		HR 22
00006	LD	TR 1
00007	AND	255 04
00008	MOV	#0001
		HR 22
00009	NEG	HR 21
		HR 21

Для следующего случая, содержимое LR 20 (#7A03) и CY вычитаются из IR 001 (#F8C5). Результат сохраняется в HR 21 и содержание HR 22 (#0000) указывает, что результат положителен.

Если бы результат был отрицательным, CY установился бы в 1, в HR занесся бы #0001 и результат был бы преобразован в дополнение до 2.

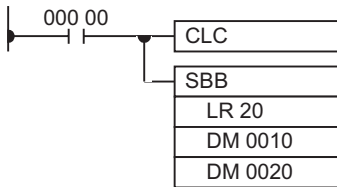


**Замечание** Могут устанавливаться также флаги UF и OF, но их можно игнорировать, поскольку они нужны только при вычитании двоичных данных со знаком.

**Пример 2: Числа со знаком**

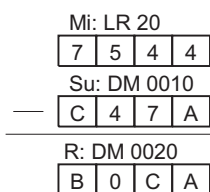
В данном примере SBB используется для вычитания одного 16-битового двоичного со знаком из другого. (Дополнение до 2 служит для выражения отрицательного значения).

Диапазон 16-битового двоичного со знаком -32 768 (8000)..+32 767 (7FFF). Флаг переполнения + (OF: SR 25404) включается в 1, если результат превышает +32 767 (7FFF) и флаг переполнения - (UF: SR 25405) включается в 1, если результат становится меньше -32 768 (8000).

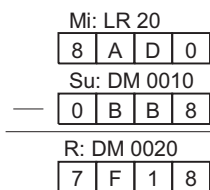


Адрес	Инструкция	Операнд
00000	LD	000 00
00001	CLC	
00002	SBB	LR 20
		DM 0010
		DM 0020

Для следующего случая  $30\ 020 - (-15\ 238) = 45\ 258$  ( $7544 - C47A = 60\ CA$ ). Флаг OF будет = 1 для указания того, что результат превышает верхнюю границу 16-битового числа со знаком. (Т. е. результат положителен, превышающий 32 767 (7FFF), а не отрицательный, выраженный двоичным значением со знаком).



В следующем случае  $-30\ 000 - 3\ 000 = -33\ 000$  ( $8AD0 - 0BB8 = 7F18$ ). Флаг UF = 1, что показывает, что результат ниже нижней границы. (Т. е. результат отрицателен, меньше -32 768 (8000), а не положительный, выраженный двоичным значением со знаком).

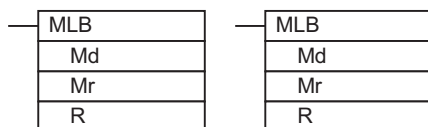


Абсолютное значение истинного результата ( $80E8 = 33\ 000$ ) можно получить, взяв дополнение до 2 числа 7F18 командой NEG.

**Замечание** Состояние флага CY можно игнорировать, поскольку оно нужно только при работе с 16-ричными числами без знака.

### 5.21.3 MLB - Умножение двоичных чисел

#### Обозначение на ПКС



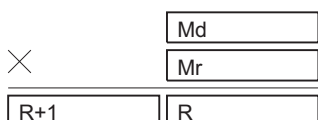
Контроллер	Номер инструкции Nr	Операнды		
		Md множимое	Mr множитель	R результат
C200H□-CPU□□-E	52	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	052	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

#### Ограничения

R и R+1 должны лежать в одной области данных.

#### Описание

Когда условие исполнения = 0, MLB не выполняется. Когда условие исполнения = 1, MLB умножает содержимое MR на содержимое MD и посылает 4 младшие цифры результата в R и 4 старшие цифры в R+1.



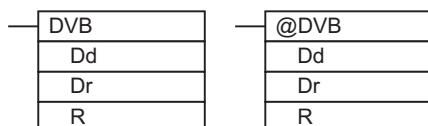
**Внимание!** MLB нельзя использовать для умножения двоичных данных со знаком. Вместо этого используйте MBS. Подробности см. 5.20.7.

#### Флаги

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- EQ:** 1, когда результат равен нулю.
- N:** 1, когда бит 15 в R+1 = 1.

### 5.21.4 DVB - Деление двоичных чисел

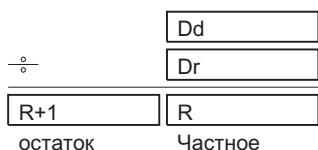
#### Обозначение на ПКС



Контроллер	Номер инструкции Nr	Операнды		
		Dd делимое	Dr делитель	R первое слово результата
C200H□-CPU□□-E	53	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	053	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

#### Описание

Когда условие исполнения = 0, DVB не выполняется. Когда условие исполнения = 1, DVB делит содержание DD на содержание DR и посылает результат в R и R+1: частное в R и остаток в R+1.





**Предосторожности**

DVB нельзя использовать для деления двоичных данных со знаком. Вместо этого используйте DBS. Подробности см. 5.20.9.

**Флаги**

- ER:** Содержание DR = 0.  
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- EQ:** 1, когда результат равен нулю.
- N:** 1, когда бит 15 в R = 1.

**5.21.5 ADBL - Сложение двоичных чисел двойной длины**

**Обозначение на ПКС**

ADBL	@ADBL
Au	Au
Ad	Ad
R	R

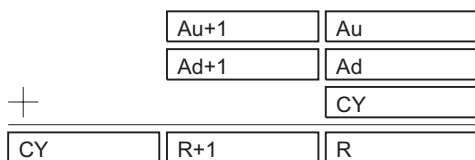
Контроллер	Номер инструкции №	Операнды		
		Au первое слово первого слагаемого	Ad первое слово второго слагаемого	R результат
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	480	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

**Ограничения**

AU и AU+1 должны лежать в одной области памяти. (То же касается AD и AD+1, R и R+1).

**Описание**

Когда условие исполнение = 0, ADBL не выполняется. Когда условие исполнения = 1, ADBL складывает 8-разрядное содержимое AU+1 и AU, 8-разрядное содержимое в AD и AD+1 и CY и заносит результат в R и R+1. CY установится в 1, когда результат больше FFFF FFFF.



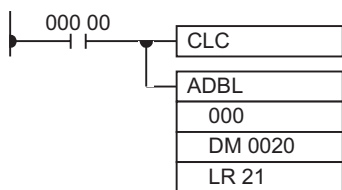
ADBL можно также использовать для сложения двоичных чисел со знаком. Флаги переполнение + и переполнение - (SR 25404 и SR 25405) указывают, когда результат перешел за верхнюю или нижнюю границу 32-битного двоичного числа со знаком. Подробности см. 3.2.

**Флаги**

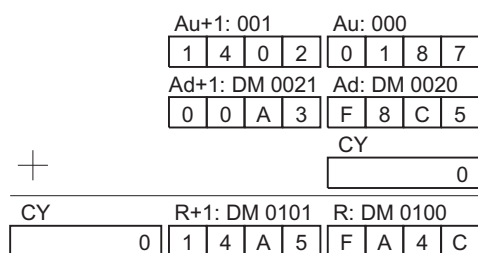
- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- CY:** 1, когда результат больше FFFF FFFF.
- EQ:** 1, когда результат равен нулю.
- OF:** 1, когда результат больше +2 147 483 647 (7FFF FFFF)
- UF:** 1, когда результат меньше - 2 147 483 648 (8000 0000)
- N:** 1, когда бит 15 в R = 1.

**Пример 1: числа без знака**

Данный пример показывает сложение 8-разрядных чисел, причем CY (SR 25504) используется для того, чтобы указать состояние 9-го числа.



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	CLC	
00002	ADBL	000
		DM 0020
		LR 21

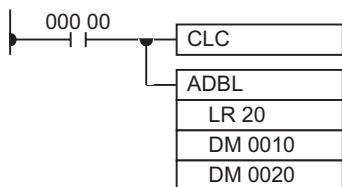


**Замечание** Состояние флагов UF и OF, можно игнорировать, поскольку они нужны только при сложении двоичных данных со знаком.

**Пример 2: Числа со знаком**

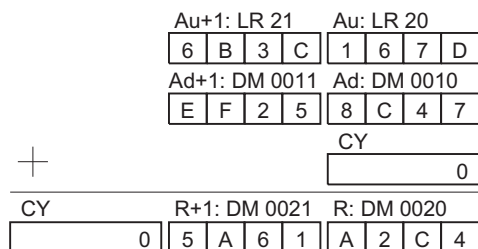
В следующем примере ADBL служит для сложения двух 32-битовых двоичных чисел со знаком и выдачи результата в R и R+1. (Дополнение до 2 служит для указания отрицательного значения).

Диапазон 32-битовых двоичных чисел со знаком: -2 147 483 648 (8000 0000)..+2 147 483 647 (7FFF FFFF). Флаг переполнения + (OF: SR 25404) включается в 1, когда результат превышает +2 147 483 647 (7FFF FFFF). Флаг переполнения - (UF: SR 25405) включается в 1, когда результат меньше -2 147 483 648 (8000 0000).



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	CLC	
00002	ADBL	LR 20
		DM 0010
		DM 0020

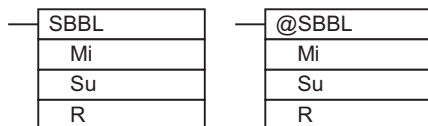
В следующем случае, 1 799 100 099 + (-282 751 929) = 1 516 348 100 (6B3C167D + EF258C47 = 5A61A2C4). Ни OF, ни UF не включаются в 1.



**Замечание** Состояние флага CY можно игнорировать, поскольку они нужны только при сложении двоичных данных без знака.

### 5.21.6 SBBL - Вычитание двоично-десятичных чисел двойной длины

#### Обозначение на ПКС



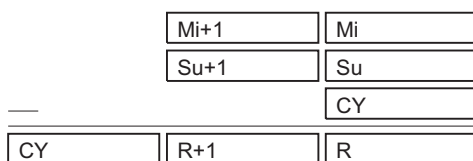
Контроллер	Номер инструкции	Операнды		
		Mi	Su	R
	Nr	первое слово уменьшаемого	первое слово вычитаемого	результат
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	481	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

#### Ограничения

MI и MI+1 должны лежать в одной области памяти. (То же касается SU и SU+1, R и R+1).

#### Описание

Когда условие исполнения = 0, SBBL не выполняется. Когда условие исполнения = 1, SBBL вычитает CY и 8-разрядное содержимое вычитаемого SU, SU+1 из 8-разрядного уменьшаемого MI, MI+1 и выдает результат в R, R+1. Если результат отрицательный, CY = 1 и в R пересылается дополнение до 2 фактического значения. Для преобразования дополнения до 2 в истинный результат используйте команду NEGL.



SBBL можно также использовать для вычитания двоичных чисел со знаком. Флаги переполнение + и переполнение - (SR 25404 и SR 25405) указывают, что результат перешел за верхнюю или нижнюю границу 32-разрядного двоичного числа со знаком.

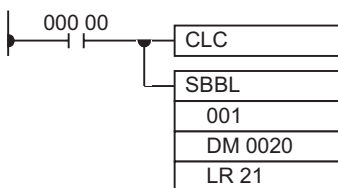
Подробности о двоичных числах со знаком см. 3.2.

#### Флаги

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- CY:** 1, когда результат больше FFFF FFFF.
- EQ:** 1, когда результат равен нулю.
- OF:** 1, когда результат больше +2 147 483 647 (7FFF FFFF)
- UF:** 1, когда результат меньше - 2 147 483 648 (8000 0000)
- N:** 1, когда бит 15 в R+1 = 1.

#### Пример 1: Числа без знака

В данном примере 8-разрядное число в IR 002 и IR 001 вычитается из 8-разрядного числа в DM 0021 и DM 0020, а результат заносится в LR 22 и LR 21. Если результат отрицателен, CY (SR 25504) = 1



Адрес	Инструкция	Операнд
00000	LD	000 00

## 5.21 Команды вычислений с двоичными числами

Адрес	Инструкция	Операнд
00001	CLC	
00002	SBBL	001
		DM 0020
		LR 21

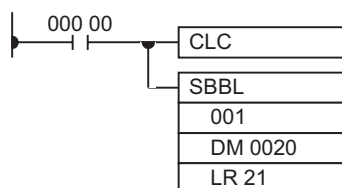
Mi+1: 002	Mi: 001	
1 4 0 2	0 1 8 7	
Md+1: DM0021	Md: DM 0020	
0 0 A 3	F 8 C 5	
	CY	
	0	
CY	R+1: LR 22	R: LR 21
0	1 3 5 E	0 8 C 2

**Замечание** Состояние флагов UF и OF, можно игнорировать, поскольку они нужны только при сложении двоичных данных со знаком.

### Пример 2: Числа со знаком

В следующем примере SBBL служит для вычитания 32-битовых двоичного числа из другого и занесения результата в R и R+1.

Диапазон 32-битовых двоичных чисел со знаком: -2 147 483 648 (8000 0000)..+2 147 483 647 (7FFF FFFF). Флаг переполнения + (OF: SR 25404) включается в 1, когда результат превышает +2 147 483 647 (7FFF FFFF). Флаг переполнения - (UF: SR 25405) включается в 1, когда результат меньше -2 147 483 648 (8000 0000).



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	CLC	
00002	SBBL	001
		DM 0020
		LR 21

В следующем случае,  $1\ 799\ 100\ 099 - (-282\ 751\ 929) = 2\ 081\ 851\ 958$  (6B3C167D - (EF258C47 - 1 0000 0000) = 7C16 8A36). Ни OF, ни UF не включаются в 1.

Mi+1: 001	Mi: 000	
6 B 3 C	1 6 7 D	
Md+1: DM0021	Md: DM 0020	
E F 2 5	8 C 4 7	
	CY	
	0	
CY	R+1: LR 22	R: LR 21
0	7 C 1 6	8 A 3 6

**Замечание** Состояние флага CY можно игнорировать, поскольку они нужны только при сложении двоичных данных без знака.

5.21.7 MBS - Умножение двоичных чисел со знаком

Обозначение на ПКС

MBS	@MBS
Md	Md
Mr	Mr
R	R

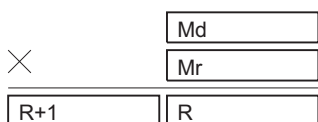
Контроллер	Номер инструкции	Операнды		
		Md	Mr	R
	Nr	первый множитель	второй множитель	результат
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	484	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

Ограничения

R и R+1 должны лежать в одной области данных.

Описание

MBS умножает двоичное содержимое со знаком двух слов и выдает 8-разрядный двоичный результат со знаком в R+1 и R. 4 младшие цифры результата заносятся в R и 4 старшие цифры в R+1. Подробности см. 3.2.



Флаги

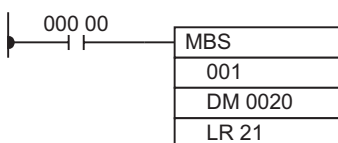
**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

**EQ:** 1, когда результат равен 0000 0000, в противном случае = 1.

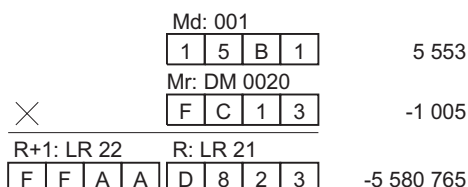
**N:** 1, когда бит 15 R+1 = 1.

Пример

В следующем примере MBS используется для умножения двоичного со знаком содержимого IR 001 на двоичное со знаком содержимое DM 0020 и выдает результат в LR 21 и LR 22.



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	MSB	001
		DM 0020
		LR 21





### 5.21.9 DBS - Деление двоичных чисел со знаком

#### Обозначение на ПКС

DBS	@DBS
Dd	Dd
Dr	Dr
R	R

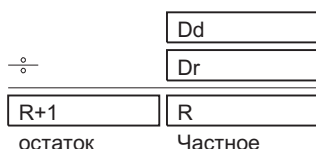
Контроллер	Номер инструкции	Операнды		
		Md	Mr	R
	Nr	делимое	делитель	первое слово результата
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	482	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

#### Ограничения

R и R+1 должны лежать в одной области данных.

#### Описание

DBS делит двоичное число со знаком DD на двоичное число со знаком DR и посылает 8-разрядный результат в R и R+1: частное в R и остаток в R+1. Подробности см. 3.2.



#### Флаги

**ER:** Содержание DR = 0.

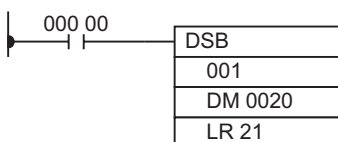
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

**EQ:** 1, когда содержание R (частное) равно нулю; в противном случае = 0.

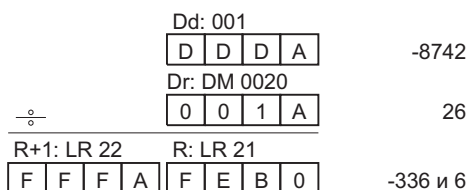
**N:** 1, когда бит 15 в R = 1.

#### Пример

В следующем примере DBS используется для деления двоичного со знаком содержимого IR 001 на двоичное со знаком содержимое DM 0020 и выдает результат в LR 21 и LR 22.



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	DBS	001
		DM 0020
		LR 21



5.21.10 DBSL - Деление двоичных чисел двойной длины со знаком

Обозначение на ПКС

DBSL	@DBSL
Dd	Dd
Dr	Dr
R	R

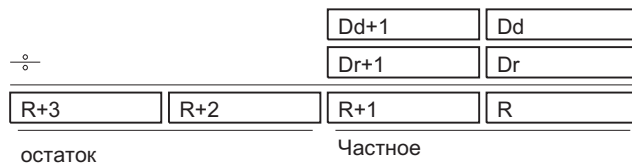
Контроллер	Номер инструкции Nr	Операнды		
		Dd делимое	Dr делитель	R результат
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	483	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

Ограничения

DD и DD+1 должны находиться в одной области данных. То же относится и к DR и DR+1, R и R+1.

Описание

DBSL делит 32-разрядное (8 цифр) двоичное число со знаком DD+1 и DD на 32-разрядное (8 цифр) двоичное число со знаком DR+1 и DR и выдает 16-разрядный двоичный результат со знаком в R+3 - R: частное в R, R+1 и остаток в R+2, R+3. Подробности о двоичных данных со знаком см. 3.2.



Флаги

**ER:** DR, DR+1 = 0.

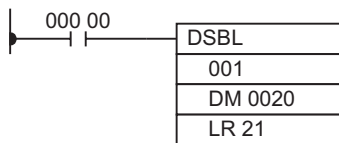
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

**EQ:** 1, когда частное (R+1 и R) равно нулю.

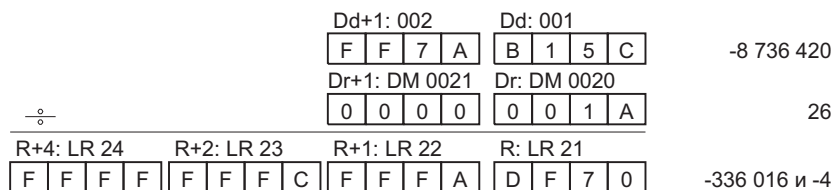
**N:** 1, когда бит 15 в R+1 = 1.

Пример

В следующем примере DBSL используется для деления двоичного со знаком содержимого IR 002 и IR 001 на двоичное со знаком содержимое DM 0021 и DM 0020 и выдает результат в LR 24..LR 21.



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	DBSL	001
		DM 0020
		LR 21





## 5.22 Специальные математические команды

### 5.22.1 MAX - Найти максимум

Обозначение на ПКС

MAX	@MAX
C	C
R1	R1
D	D

Контроллер	Номер инструкции №	Операнды		
		C	R1	D
		слово состояния	первое слово в блоке	слово приемник
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, DM, #	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	182	IR, SR, HR, AR, LR, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

#### Ограничения

N в C должно быть двоично-десятичным числом от 001 до 999.

R1 и R1+N-1 должны лежать в одной области памяти.

#### Описание

Когда условие исполнение = 0, MAX не выполняется. Когда условие исполнения = 1, MAX просматривает область памяти от R1 до R1+N-1 для того, чтобы найти адрес, содержащий максимальное значение и выдает максимальное значение в слово приемника D.

Если бит 14 слова C = 1, MAX идентифицирует адрес максимального значения в D+1.

Для области DM адрес идентифицируется иначе, чем для других областей:

- 1, 2, 3,... 1. Для области DM адрес записывается в D+1. Например, если адрес, содержащий максимум, = DM 0114, тогда #0114 запишется в D+1.
2. Для других областей данных в D+1 записывается количество адресов от начала поиска. Например, если адрес, содержащий максимум, = IR 0114 и первое слово зоны поиска = IR 014, тогда #0100 запишется в D+1.

Если бит 14 слова C = 1 и максимальное значение содержится по нескольким адресам, в D+1 будет помещен самый младший адрес.

Количество слов в зоне (N) находится в 3 младших цифрах C, которые должны быть двоично-десятичными 001.. 999.

Если бит 15 слова C = 0, данные внутри зоны считаются двоичными без знака, а когда = 1, то двоичными со знаком.



**Внимание!** Если бит 14 C = 1, значения свыше #8000 считаются отрицательными, так что результаты будут различными в зависимости от заданного типа данных. Следите за тем, чтобы был задан правильный тип.

#### Флаги

**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

Число слов, заданных в C, не двоично-десятичные числа от 001 до 999.

R1 и R1+N-1 находятся в разных областях.

**EQ:** 1, когда результат равен нулю.

**N:** 1, когда бит 15 = 1.

### 5.22.2 MIN - Найти минимум

#### Обозначение на ПКС

MIN	@MIN
C	C
R1	R1
D	D

Контроллер	Номер инструкции	Операнды		
		C	R1	D
	№г	слово состояния	первое слово в блоке	слово приемник
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, DM, #	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	183	IR, SR, HR, AR, LR, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

#### Ограничения

N в C должно быть двоично-десятичным числом от 001 до 999.

R1 и R1+N-1 должны лежать в одной области памяти.

#### Описание

Когда условие исполнения = 0, MIN не выполняется. Когда условие исполнения = 1, MIN просматривает зону памяти от R1 до R1+N-1 для того, чтобы найти адрес, содержащий минимальное значение и выдает минимальное значение в слово приемника D.

Если бит 14 слова C = 1, MIN записывает адрес минимального значения в D+1. Для области DM адрес идентифицируется иначе, чем для других областей:

- 1, 2, 3,...**
1. Для области DM в D+1 записывается абсолютный адрес. Например, если адрес, содержащий максимум, = DM 0114, тогда в D+1 запишется #0114.
  2. Для других областей данных в D+1 записывается количество адресов от начала поиска. Например, если адрес, содержащий минимум, = IR 0114 и первое слово зоны поиска = IR 014, тогда #0100 запишется в D+1.

Если бит 14 слова C = 1 и минимальное значение содержится по нескольким адресам, в D+1 будет помещен самый младший адрес.

Количество слов в зоне (N) находится в 3 младших цифрах C, которые должны быть двоично-десятичными 001.. 999.

Если бит 15 слова C = 0, данные внутри зоны считаются двоичными без знака, а когда = 1, то двоичными со знаком. Подробности о двоичных со знаком см. 3.2.



**Внимание!** Если бит 14 C = 1, значения свыше #8000 считаются отрицательными, так что результаты будут различными в зависимости от заданного типа данных. Следите за тем, чтобы был задан правильный тип.

**Флаги**

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).  
Число слов, заданных в С, не двоично-десятичные числа от 001 до 999.  
R1 и R1+N-1 находятся в разных областях.
- EQ:** 1, когда минимальный результат равен нулю (#0000).
- N:** 1, когда бит 15 = 1.

**5.22.3 AVG - Среднее значение**

**Обозначение на ПКС**

AVG	@AVG
S	S
N	N
D	D

Контроллер	Номер инструкции	Операнды		
		S	N	D
		слово источник	число циклов	
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	195	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Ограничения**

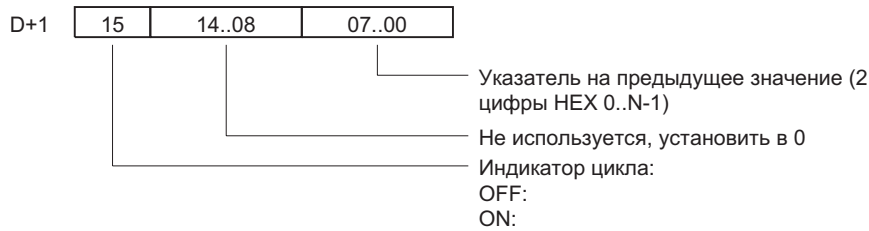
- S должно быть 16-ричным
- N должно быть двоично-десятичным числом от #0001 до #0064.
- D и D+N+1 должны лежать в одной области памяти.

**Описание**

AVG служит для вычисления среднего значения S за N циклов.  
 Когда условие исполнения = 0, AVG не выполняется.  
 Когда условие исполнения = 1, первые N-1 циклов AVG записывает значение S в D. Каждый раз при исполнении AVG предыдущее содержимое S заносится в слова D+2 до D+N+1. Первые 2 цифры D+1 инкрементируются с каждым исполнением, и действует как указатель для индикации, куда записано предыдущее значение. Бит 15 D+1 остается в 0 первые N-1 циклов.  
 При N-м цикле предыдущее значение S записывается в зону D+2.. D+N-1. Среднее значение предыдущих значений, загруженных в D+2.. D+N+1 вычисляется и записывается в D, бит 15 D+1 устанавливается в 1 и значение предыдущего указателя (первые 2 цифры D+1) сбрасывается в 0. Каждый раз при выполнении AVG предыдущее значение S затирает содержимое слова, на которое указывает указатель, и новое среднее значение вычисляется и записывается в D. Указатель будет снова сброшен по достижении N-1.  
 В таблице показаны функции слов D.. D +N+1.

D	Среднее значение (после N циклов или более)
D+1	Указатель предыдущего значения и указатель цикла.
D+2	Предыдущее значение #1
D+3	Предыдущее значение #2
..	
D+N+1	Предыдущее значение #N

Функция битов в D+1 на следующем рисунке объясняется ниже.



**Указатель предыдущего значения**

Указатель предыдущего значения указывает местоположение, куда загружено последнее S, относительно D+2, т.е. значения указателя 0 указывает на D+2, а значение 1 указывает на D+3 и т.д.

**Указатель цикла**

Указатель цикла включается в 1 после того, как AVG выполнилась N раз. В этот момент D будет содержать среднее значение D+2.. D+N+1. Среднее значение - это 4-разрядное 16-ричное число, и оно округляется до ближайшего целого.

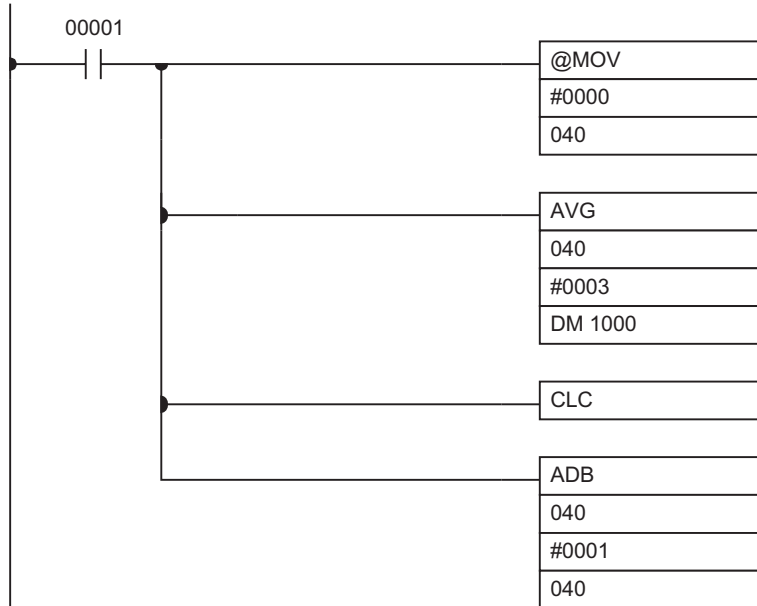
**Флаги**

**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

Один или более операндов задан некорректно.

**Пример**

В следующем примере содержимое IR 040 установлено в #0000 и затем инкрементируется каждый цикл. Для первых двух циклов ANG переносит содержимое IR 040 в DM 1002 и DM 1003. Содержимое DM 1001 также изменится (это можно использовать для подтверждения того, что результаты AVG изменились). В третий и последующие циклы AVG подсчитывает среднее значение содержимого DM 1002.. DM 1004 и заносит это значение в DM 1000.



Адрес	Инструкция	Операнд
00000	LD	000 01
00001	@MOV	#0000
		040
00002	AVG	040
		#0003
		DM 1000
00003	CLC	

## 5.22 Специальные математические команды

Адрес	Инструкция	Операнд
00004	ADB	040
		#0001
		040

	1-й цикл	2-й цикл	3-й цикл	4-й цикл	
IR 040	0000	0001	0002	0003	
DM 1000	0000	0001	0001	0002	Среднее значение
DM 1001	0001	0002	8000	8001	Указатель
DM 1002	0000	0000	0000	0003	
DM 1003	-	0001	0001	0001	3 Предыдущих значения IR 40
DM 1004	-	-	0002	0003	

### 5.22.4 SUM - Сумма

#### Обозначение на ПКС

SUM	@SUM
C	C
R1	R1
D	D

Контроллер	Номер инструкции №	Операнды		
		C	R1	D
C200H□-CPU□□-E	-	слово состояния IR, SR, HR, AR, LR, DM, #	первое слово в блоке IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	184	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

#### Ограничения

3 младших цифры C должны быть двоично-десятичными от 001 до 999.

Если бит 14 слова C = 0 (задание двоично-десятичных величин), все данные в зоне R1.. R1+N-1 должны быть двоично-десятичными.

#### Описание

Когда условие исполнение = 0, SUM не выполняется. Когда условие исполнения = 1, SUM складывает содержимое слов R1.. R1+N-1 или байт в словах R1.. R1+N/2-1 и выдает результат в слова приемника D и D+1. Можно суммировать данные в двоичном или двоично-десятичном виде и выдавать в таком же виде. Двоичные данные могут быть со знаком или без знака.

Назначение битов в слове управления показано в следующей диаграмме и объяснено далее.



**Количество объектов в зоне**

Количество объектов в зоне (N) содержится в трех младших цифрах слова С и должно быть двоично-десятичным числом от 001 до 999. Количество будет указывать на количество либо слов, либо байт в зависимости от объектов суммирования.

**Единица сложения**

Если бит 13 = 0, будут складываться слова, если бит 13 = 1, то биты.

Если заданы байты, зона может начинаться либо с младшего, либо старшего байта R1.

Если бит 12 = 0, старший байт R1 не будет складываться.

	Старший байт	Младший байт
R1	1	2
R1+1	3	4
R1+2	5	6
R1+3	7	8
..		

Если бит 12 = 0, байты будут складываться в следующем порядке: 1+2+3+4..

Если бит 12 = 1, байты будут складываться в следующем порядке: 2+3+4..

**Тип данных**

Если в слове С бит 14 = 1, а бит 15 = 0, данные в зоне считаются двоичными без знака;

Если в слове С бит 14 = 1 и бит 15 = 1, данные в зоне считаются двоичными со знаком

Если в слове С бит 14 = 0, данные в зоне считаются двоично-десятичными независимо от состояния бита 15.

**Флаги**

**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

R1 и R1+N-1 находятся в разных областях.

Количество объектов в С не является двоично-десятичным числом от 001 до 999.

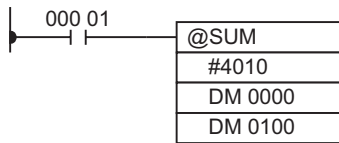
Данные, подлежащие суммированию, не являются двоично-десятичными числами, когда задано сложение двоично-десятичных чисел.

**EQ:** 1, когда результат равен нулю.

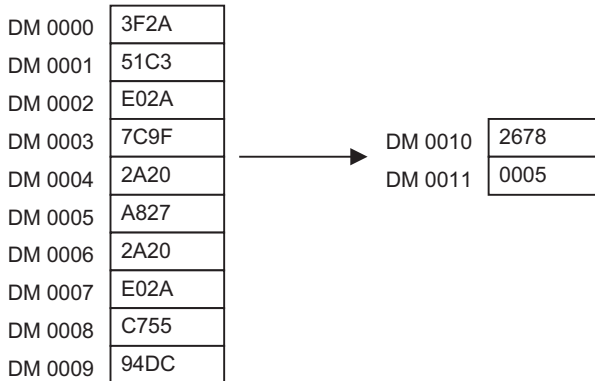
**N:** 1, когда бит 15 D = 1.

**Пример**

В следующем примере двоично-десятичное содержимое 10 слов от DM 0000 до DM 0009 складываются, когда IR 00001 = 1, и результат заносится в DM 0010 и DM 0011.

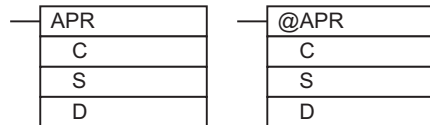


Адрес	Инструкция	Операнд
00000	LD	000 01
00001	@SUM	#4010
		DM 0000
		DM 0100



5.22.5 APR - Математические вычисления

Обозначение на ПКС



Контроллер	Номер инструкции №	Операнды		
		C	S	D
C200H-CPU-E	69	слово состояния	входные данные	результат
C200H-CPU-ZE	069	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
		IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

Ограничения

Для тригонометрических функций S должно быть двоично-десятичным числом в диапазоне 0000.. 0900 (0<sup>0</sup>..090<sup>0</sup>).

Описание

Когда условие исполнение = 0, APR не выполняется. Когда условие исполнения = 1, операции APR зависят от слова управления C.

Если C = #0000 или #0001, APR вычисляет SIN или COS. Двоично-десятичное значение S задает угол в одной десятой градуса.

Если C является адресом, APR вычисляет f(x) функцию, введенную ранее, имеющую начало в слове C.

Функция - это серия отрезков прямой (которыми аппроксимируется кривая), заданная оператором. Двоично-десятичное или 16-ричное значение S задает x.

Флаги

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- Для тригонометрических функций x 0900. (x - содержимое S).
- В S задана константа, отличная от #0000 и #0001.
- Данные линейной аппроксимации ненадежны.

**EQ:** 1, когда результат равен нулю.

**N:** 1, когда бит 15 слова D = 1..

**Примеры:**

**Синус**

В следующем примере показано использование функции синуса APR для вычисления синуса  $30^\circ$ . Функция синуса задана, когда C = #0000.



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	APR	#0000
		DM 0000
		DM 0100

Входные данные, x

S: DM 0000			
0	10 <sup>1</sup>	10 <sup>0</sup>	10 <sup>-1</sup>
0	3	0	0

Введите данные, не превышающие #0900 в BCD

Результат

D: DM 0100			
10 <sup>-1</sup>	10 <sup>-2</sup>	10 <sup>-3</sup>	10 <sup>-4</sup>
5	0	0	0

Результат имеет 4 значащие цифры, пятая и далее игнорируются. Результат для Sin(90) будет 0.9999 а не 1

**Косинус**

В следующем примере показано использование функции косинуса APR для вычисления косинуса  $30^\circ$ . Функция косинуса задана, когда C = #0001.



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	APR	#0000
		DM 0010
		DM 0110

Входные данные, x

S: DM 0000			
0	10 <sup>1</sup>	10 <sup>0</sup>	10 <sup>-1</sup>
0	3	0	0

Введите данные, не превышающие #0900 в BCD

Результат

D: DM 0100			
10 <sup>-1</sup>	10 <sup>-2</sup>	10 <sup>-3</sup>	10 <sup>-4</sup>
8	6	6	0

Результат имеет 4 значащие цифры, пятая и далее игнорируются. Результат для Cos(0) будет 0.9999 а не 1

**Линейная аппроксимация**

Линейная аппроксимация APR задана, когда в C установлен адрес памяти. Слово C является первым словом блока памяти, содержащего параметры линейной аппроксимации.

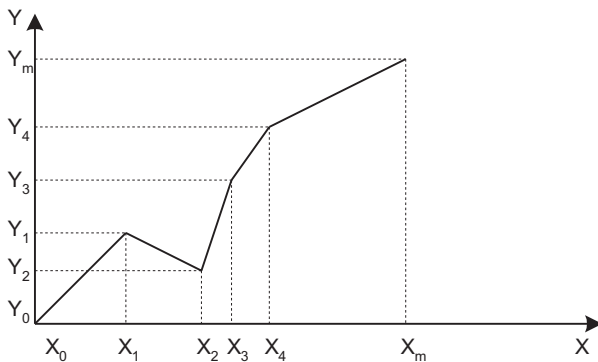


## 5.22 Специальные математические команды

Содержимое слова С задает количество отрезков линий в аппроксимации, и вид задания ввода и вывода - двоично-десятичный или двоичный. Биты 00 - 07 содержат количество сегментов минус 1, как двоичные данные. Биты 14 и 15 определяют, соответственно, формы ввода и вывода: 0 - двоично-десятичная, 1 - двоичная.

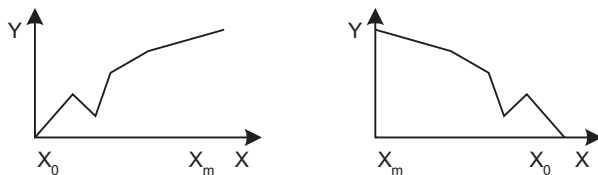


Введите координаты  $m+1$  конечных точек, которые определяют  $m$  отрезков, как показано в таблице. Все координаты вводите в двоичной форме. Всегда вводите координаты в направлении с наименьшего значения  $X$  ( $X_1$ ) до наибольшего ( $X_m$ ).  $X_0 = 0000$ , и его вводить не нужно.



Слово	Координата
C+1	$X_m$
C+2	$Y_0$
C+3	$X_1$
C+4	$Y_1$
C+5	$X_2$
C+6	$Y_2$
..	
C+(2m+1)	$X_m$
C+(2m+2)	$Y_m$

Если бит 13 слова С = 1, график будет зеркально перевернут слева направо, как показано на следующих рисунках.

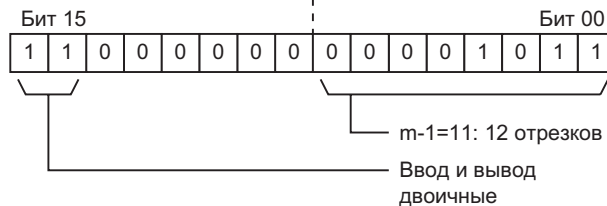


Следующий пример демонстрирует конструкцию линейной аппроксимации с 12 отрезками прямых. Блок данных, как и положено, непрерывен от DM 0000 - DM 0026 (от С до  $C+(2 \times 12=24)$ ). Входные данные берутся из IR 010, и результат выдается в IR 011.

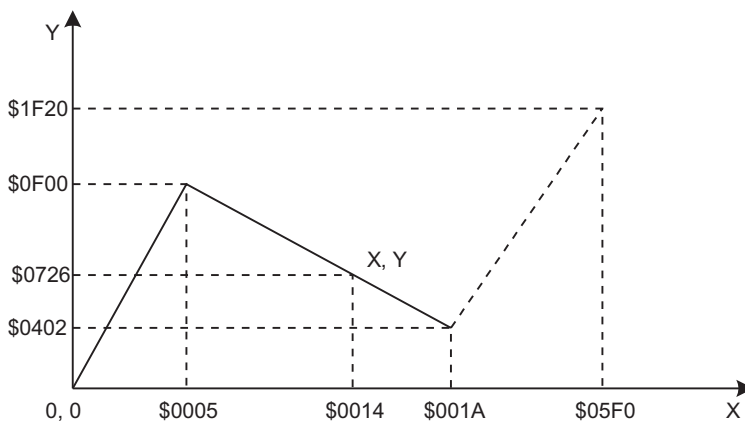


Адрес	Инструкция	Операнд
00000	LD	000 00
00001	APR	DM 0000
		010
		011

	Содержание	Координата
DM 0000	\$C00B	
DM 0001	\$C5F0	X <sub>12</sub>
DM 0002	\$0000	Y <sub>0</sub>
DM 0003	\$0005	X <sub>1</sub>
DM 0004	\$0F00	Y <sub>1</sub>
DM 0005	\$001A	X <sub>2</sub>
DM 0006	\$0402	Y <sub>2</sub>
DM 0025	\$05F0	X <sub>12</sub>
DM 0026	\$1F20	Y <sub>12</sub>

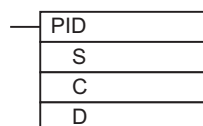


В данном случае слово входных данных, IR 010, содержит #0014 и  $f(0014) = \#0726$  выдается в R, IR 011.



### 5.22.6 PID - PID-регулирование

Обозначение на ПКС



## 5.22 Специальные математические команды

Контроллер	Номер инструкции	№	Операнды		
			S	C	D
			входные данные	первое слово блока параметров	выходные данные
C200H□-CPU□□-E	-		IR, SR, HR, AR, LR, DM	IR, SR, HR, LR, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	190		IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, LR, DM, EM	IR, SR, HR, AR, LR, DM, EM

### Ограничения

C и C+32 должны быть в одной области.

**Замечание** Не программируйте PID в далее перечисленных ситуациях. Это может привести к непредсказуемым результатам.

- в программах прерываний;
- в подпрограммах;
- между IL и ILC;
- между JMP и JME;
- в секциях программы STEP и STEP.

### Описание

PID осуществляет пропорционально-интегрально-дифференциальное управление согласно заданным параметрам. Команда берет заданный входной диапазон (двоичное число) из содержимого входного слова S и выполняет PID-регулирование согласно заданным параметрам. Результаты (выходное значение регулирования) затем выдаются в выходное слово D.

Параметры PID-регулирования находятся в словах C.. C+32. Параметры PID располагаются следующим образом.

Слово	15.. 12	11.. 8	7.. 4	3.. 0
C	Задание (SV)			
C+1	Ширина пропорционального диапазона (P)			
C+2	T <sub>ik</sub> = время интегрального регулирования T <sub>1</sub> /период выборки _ (См. прим. 1)			
C+3	T <sub>dk</sub> = время дифференциального регулирования T <sub>d</sub> /период выборки _ (См. прим. 1)			
C+4	Период выборки _			
C+5	2-PID параметр ( ) (См. прим. 2)			Направление PID (прямое/обратное)
C+6	0	Входной диапазон	Единица времени	Выходной диапазон
C+7.. C+32	Рабочая область (Недоступна непосредственно из программы).			

- Замечание**
- Фактическое время интегрирования и дифференцирования вычисляются с помощью значений, заданных в C+2 и C+3 и дискреты времени в C+6.
  - Задание 2-PID параметра ( ) в 000 дает 0.65, нормальное значение.

### Значения параметров

Параметр	Содержание	Диапазон задания
Задание (SV)	Конечное значение управляемого процесса.	Двоичное значение (такое же количество бит, как и для диапазона входа).
Ширина пропорционального диапазона	Параметр пропорционального управления. Выражает соотношение "ширина диапазона пропорционального регулирования" / "общий диапазон регулирования".	4-разрядное двоично-десятичное число от 0001 до 9999. ( 0.1%.. 999.9%, дискрета 0.1%)
T <sub>ik</sub>	Константа, выражающая интенсивность интегрального регулирования. По мере увеличения значения это величины, степень интегрального регулирования уменьшается. Параметр дискреты времени определяет метод задания.	4-разрядное двоично-десятичное число от 0001 до 8191. (9999 = нет интегрального регулирования). 1x8191x когда единица времени = 0 или 1 0.1.. 819.1 с, когда единица времени = 8 0.1.. 81.91 с, когда единица времени = 9

## 5.22 Специальные математические команды

Параметр	Содержание	Диапазон задания
Tdk	Константа, выражающая интенсивность дифференциального регулирования. По мере увеличения значения этой величины, интенсивность дифференциального регулирования увеличивается. Параметр дискреты времени определяет метод задания.	4-разрядное двоично-десятичное число от 0001 до 8191. (0000 = нет дифференциального регулирования). 1x8191x когда единица времени = 0 или 1 0.1.. 819.1 с, когда единица времени = 8 0.1.. 81.91 с, когда единица времени = 9
Период выборки	Задаёт период PID- регулирования.	4-разрядное двоично-десятичное число от 0001 до 9999. (0.01.. 99.99 с, дискрета 0.01 с).
Указатель: прямое/ инверсное регулирование	Параметр определяет направление пропорционального регулирования.	0 - инверсное; 1 - прямое. (1-разрядное двоично-десятичное число)
2-PID параметр _	Коэффициент входного фильтра. Нормальное значение 0.65 (задание 000). Эффективность фильтра уменьшается по мере приближения коэффициента к 0.	000: _ = 0.65 Задание 100.. 999 значит, что значение двух правых (младших цифр) задано _ = 0.00.. _ = 0.99. (3-разрядное двоично-десятичное число)
Входной диапазон	Число битов данных входа	0: 8 бит 1: 8 бит 2: 10 бит 3: 11 бит 4: 12 бит (BCD) 5: 13 бит 6: 14 бит 7: 15 бит 8: 16 бит (1 цифра BCD)
Дискрета времени	Определяет метод задания параметров интегрирования/ дифференцирования	0, 1, 8 или 9 (1 двоично-десятичная цифра) 0 или 1: задание времени 8: задание времени (дискрета 100 мс) 9: задание времени (дискрета 10 мс)
Выходной диапазон.	Число битов данных выхода	Те же значения, что и для входного диапазона.

### PID-регулирование

#### Условия исполнения = 0

Все заданные параметры сохраняются. Поскольку условия исполнения = 0, выход регулирования можно записать в выходное слово (D) для осуществления ручного управления.

#### Передний фронт условия исполнения

Рабочая область инициализируется, в соответствии с заданными параметрами PID, и операция PID-регулирования начинается. В начале операции резкие изменения выхода регулирования нет во избежание негативного воздействия на управляемую систему (бестолчковое управление).

При изменении параметров PID они впервые вступают в действие при изменении условий исполнения из 0 в 1.

#### Условия исполнения = 1

Выполняется PID-регулирование с интервалами, основанными на периоде выборки согласно заданным параметрам PID.

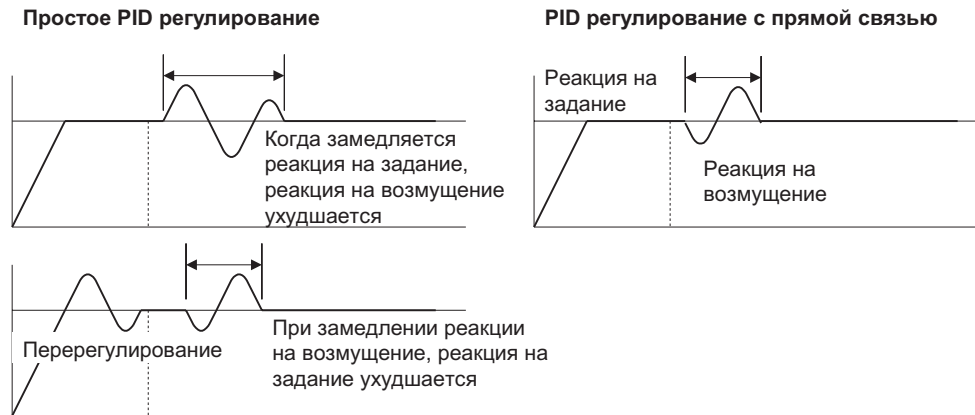
#### Период выборки и синхронизация исполнения PID

Период выборки - это временной интервал для снятия данных измерения для выполнения операции PID. Команда PID, однако, выполняется в соответствии со временем выборки ЦПУ, так что могут быть случаи, когда период выборки превышен. В таких случаях временной интервал до следующей выборки сокращается.

### Метод PID-регулирования

Операции PID-регулирования выполняются методом управления прямой связи (две степени свободы).

Когда перерегулирование предотвращается простым PID-регулированием, стабилизация возмущений замедляется (1). С другой стороны, если стабилизацию возмущений ускорить, появится перерегулирование и реакция на задание замедлится (2). При PID-регулировании с прямой связью нет перерегулирований, и реакция в направлении к заданию и стабилизация возмущений ускоряется (3).



## Регулирование

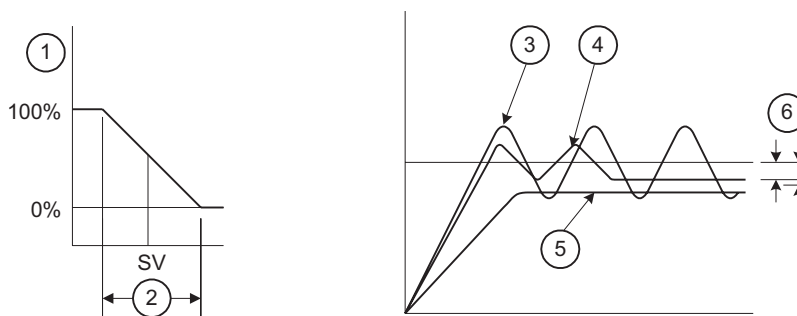
### Пропорциональное регулирование (P)

Пропорциональное регулирование - это регулирование, при котором пропорциональный диапазон устанавливается в соответствии с заданием, и внутри этого диапазона выход регулирования пропорционален отклонению. Если текущее значение меньше пропорционального диапазона, выход регулирования = 100 %. Если внутри пропорционального диапазона выход регулирования пропорционален отклонению и постепенно уменьшается, пока задание и текущее значение не будут соответствовать друг другу (т. е. пока отклонение не будет = 0), выход регулирования возвратится к предыдущему значению (прямое регулирование).

Пропорциональный диапазон выражен в процентах относительно полного диапазону входа. При пропорциональном регулировании появляется рассогласование (остаточное смещение) и рассогласование уменьшается путем сужения пропорционального диапазона. Если он сделался слишком малым, начнутся колебания.

**Пропорциональное регулирование (прямое регулирование)**

**Настройка пропорционального диапазона**



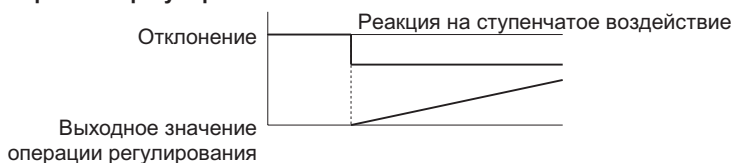
1. Выходное значение операции регулирования
2. Пропорциональный диапазон
3. Пропорциональный диапазон слишком узок (происходят колебания)
4. Пропорциональный диапазон задан правильно
5. Пропорциональный диапазон слишком широк (большое рассогласование)
6. Рассогласование

### Интегральное регулирование (I)

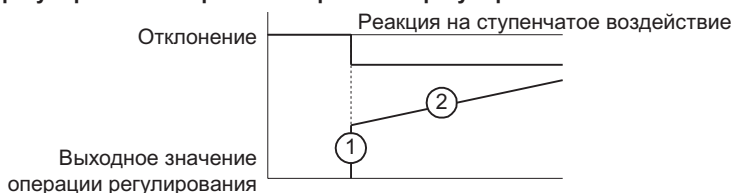
Комбинирование интегрального регулирования с пропорциональным сокращает рассогласование (соответственно прошедшему времени). Интенсивность интегрального регулирования определяется временем интегрирования (это время, требуемое для того, чтобы степень интегрального регулирования достигла уровня пропорционального управления по отношению к шаговому отклонению, как показано на следующем рисунке).

Чем меньше время интегрального регулирования, тем сильнее будет коррекция при интегральном регулировании. Если интегральное время слишком коротко, коррекция будет слишком сильной и вызовет колебания.

#### Интегральное регулирование



#### PI регулирование и время интегрального регулирования



1. I: интегральная составляющая
2. P: пропорциональная составляющая

#### Дифференциальное регулирование (D)

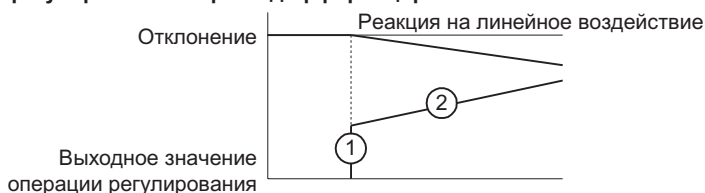
И пропорциональное регулирование, и интегральное, оба производят коррекцию в соответствии с результатами регулирования, так что неизбежно будет задержка времени. Дифференциальное регулирование компенсирует этот недостаток. При реакции на резкое возмущение оно производит регулирование с большой интенсивностью и быстро восстанавливает исходное состояние. Коррекция производится со степенью, пропорциональной наклону (коэффициенту отклонения), вызванному отклонением.

Интенсивность дифференциального управления определяется временем дифференцирования (это время, требуемое для того, чтобы степень дифференциального регулирования достигла уровня пропорционального регулирования по отношению к шаговому отклонению, как показано на следующем рисунке). Чем больше время дифференциального регулирования, тем сильнее будет коррекция, производимая дифференциальным регулированием.

#### Дифференциальное регулирование



#### PD регулирование и время дифференцирования

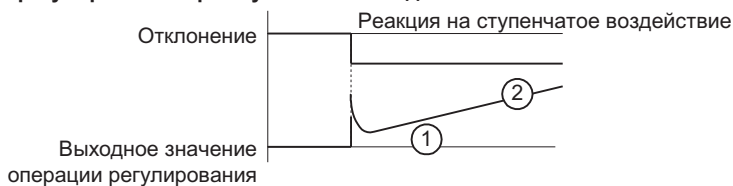


1. D: дифференциальная составляющая
2. P: пропорциональная составляющая

#### PID-регулирование (пропорционально-интегрально-дифференциальное)

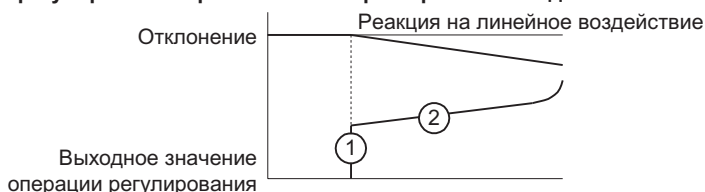
PID-регулирование комбинирует пропорциональное регулирование (P), интегральное (I) и дифференциальное (D). Оно обеспечивает отличные результаты регулирования даже при управлении объектами с "мертвым временем". Пропорциональное регулирование обеспечивает плавное регулирование без колебаний, интегральное автоматически корректирует отклонения, а дифференциальное ускоряет реакцию на возмущения.

**PID регулирование при ступенчатом воздействии**



- 1. D: дифференциальная составляющая
- 2. P: пропорциональная составляющая
- 3. I: интегральная составляющая

**PID регулирование при линейной характеристике входа**



- 1. D: дифференциальная составляющая
- 2. P: пропорциональная составляющая
- 3. I: интегральная составляющая

**Направление регулирования**

При использовании PID- регулирования выбирайте одно из следующих двух направлений. При обоих направлениях интенсивность регулирования возрастает при увеличении разности между заданным и текущим значениями.

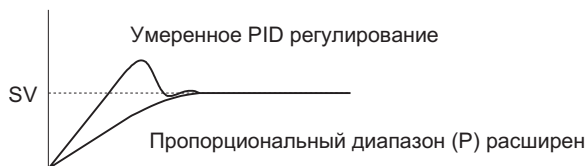
Прямое регулирование: Интенсивность регулирования возрастает, когда текущее значение больше заданного.

Обратное регулирование: Интенсивность регулирования возрастает, когда текущее значение меньше заданного.

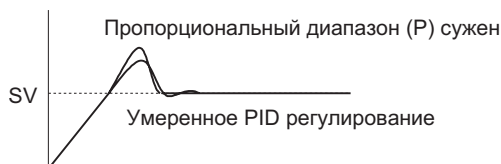
**Настройка параметров PID-регулирования**

Общее соотношение между параметрами PID и состоянием регулирования показано ниже.

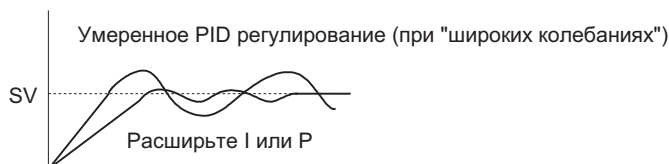
- Когда неважно время стабилизации, но важно отсутствие перерегулирования, тогда рекомендуем расширить пропорциональный диапазон.



- Когда неважно перерегулирование, но желательно быстро стабилизировать регулирование, тогда рекомендуем сузить пропорциональный диапазон. Если пропорциональный диапазон сужен слишком сильно, могут начаться колебания.



- Когда наличествуют широкие колебания, или когда регулирование затруднено перерегулированием или недостаточным регулированием, причина, скорее всего - интегральное регулирование слишком сильно. Колебания будут уменьшены, если время интегрального регулирования увеличивается или пропорциональный диапазон расширяется.



- Если период коротко и происходят колебания, вероятно, что реакция системы слишком быстрая, и дифференциальное регулирование слишком сильно. В данном случае уменьшите степень дифференциального регулирования.



### Флаги

**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

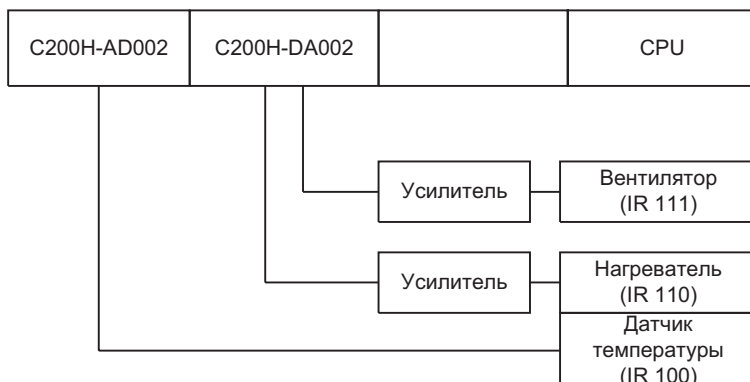
Задание параметра PID выходит за допустимый диапазон.

PID-регулирование было проведено, но время цикла было в два раза больше периода выборки. PID будет выполняться при этой ошибке, даже если когда ER (SR 25503) = 1.

**СУ:** 1, когда PID завершилась.

### Пример

Данный пример показывает программу PID-регулирования, использующую команду PID.



**Замечание** Двигатели и нагреватели нельзя подключать непосредственно к блоку аналогового выхода. Необходим усилитель (т. е. цепь усиления тока).

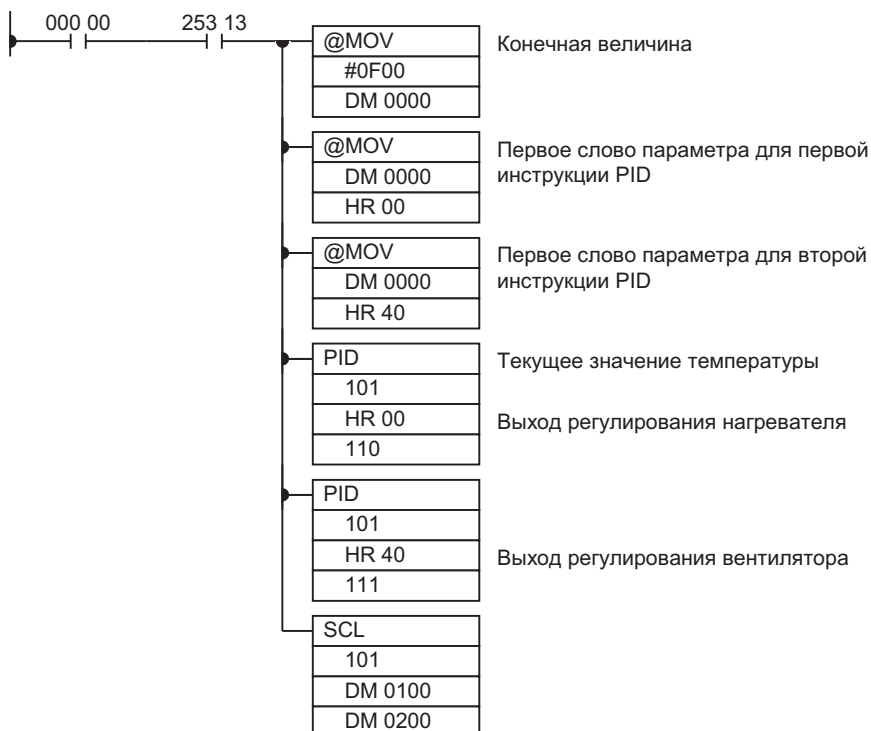
### Создание программы

Для создания программы пользуйтесь следующим алгоритмом.

- Замечание**
1. Задайте заданное значение в DM 0000 (двоичное число в диапазоне 0000..0FFF).
  2. Введите текущее значение из температурного датчика (двоичное число в диапазоне 0000..0FFF) в биты 0.. 11 слова 101.
  3. Выдайте степень регулирования нагревателя в биты 0.. 11 слова 110 с помощью первой команды PID следующей программы.
  4. Выдайте степень регулирования вентилятора в биты 0.. 11 слова 11 с помощью второй команды PID следующей программы.
  5. Преобразуйте текущее значение датчика температуры (двоичное число в диапазоне 000..FFF) в температуру (0000 C<sup>0..0200 C<sub>0</sub></sup>) командой SCL и выдайте ее в DM 0200.



Программа



**Замечание** При использовании команд PID или SCL предварительно задавайте параметры с периферийного устройства (программатора или LSS).

**Нагреватель**

HR 00	(DM 0000)	Конечное значение HR
HR 01	0080	Пропорциональный диапазон
HR 02	0200	Время интегрирования / период выборки
HR 03	0100	Время дифференцирования / период выборки
HR 04	0001	Период выборки
HR 05	0000	Прямое / обратное регулирование, параметры PID
HR 06	0404	Диапазон входов/выходов и дискрета времени задания

**Параметры SCL**

DM 0100	0000
DM 0101	0000
DM 0102	0200
DM 0103	0FFF

**Вентилятор**

HR 40	(DM 0000)
HR 41	0060
HR 42	0150
HR 43	0100
HR 44	0001
HR 45	0001
HR 46	0404

## 5.23 Логические команды

Логические команды - COM, ANDW, ORW, XORW и XNRW - производят логические операции над словами.

### 5.23.1 COM - Дополнение

Обозначение на ПКС

COM	@COM
Wd	Wd

Контроллер	Номер инструкции №	Операнды		
		Wd		
		преобразуемое слово		
C200H□-CPU□□-E	29	IR, SR, HR, AR, LR, DM		
C200H□-CPU□□-ZE	029	IR, SR, HR, AR, LR, DM, EM		

Описание

Когда условие исполнение = 0, COM не выполняется. Когда условие исполнения = 1, COM очищает в слове WD биты, находящиеся в состоянии 1 и устанавливает d 1 все биты, находящиеся в состоянии 0.

COM будет постоянно менять значение слова WD каждый цикл при условии исполнении = 1. При необходимости используйте @COM.

Пример

	Бит 15		Бит 00													
Исходное слово	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
	↓															
Дополнение	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0

Флаги

**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

**EQ:** 1, когда результат равен нулю.

**N:** 1, когда бит 15 слова WD = 1.

### 5.23.2 ANDW - Логическое И

Обозначение на ПКС

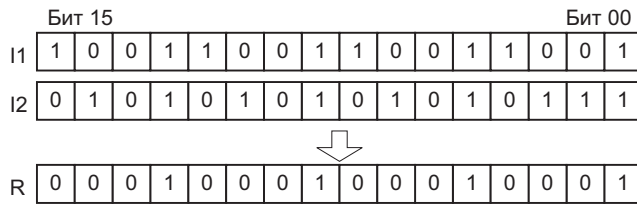
ANDW	@ANDW
I1	I1
I2	I2
R	R

Контроллер	Номер инструкции №	Операнды		
		I1	I2	R
		первое слово	второе слово	результат
C200H□-CPU□□-E	34	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	034	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

Описание

Когда условие исполнение = 0, ANDW не выполняется. Когда условие исполнения = 1, ANDW производит операцию ЛОГИЧЕСКОЕ И с содержимым I1 и I2 побитно и выдает результат в R.

**Пример**



**Флаги**

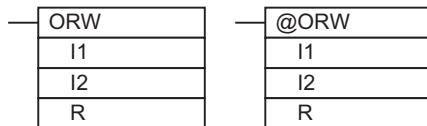
**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

**EQ:** 1, когда результат равен нулю.

**N:** 1, когда бит 15 слова R = 1.

**5.23.3 ORW - Логическое ИЛИ**

**Обозначение на ПКС**

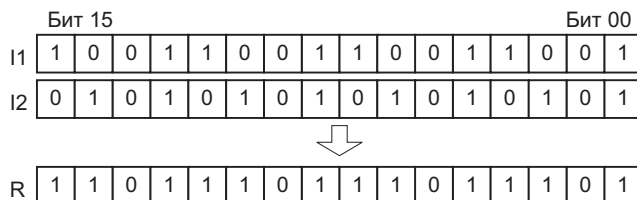


Контроллер	Номер инструкции №	Операнды		
		I1 первое слово	I2 второе слово	R результат
C200H□-CPU□□-E	35	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	035	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Описание**

Когда условие исполнение = 0, ORW не выполняется. Когда условие исполнения = 1, ORW производит операцию ЛОГИЧЕСКОЕ ИЛИ с содержимым I1 и I2 побитно и выдает результат в R.

**Пример**



**Флаги**

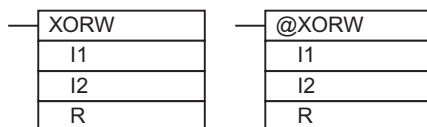
**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

**EQ:** 1, когда результат равен нулю.

**N:** 1, когда бит 15 слова R = 1.

**5.23.4 XORW - Исключающее ИЛИ**

**Обозначение на ПКС**

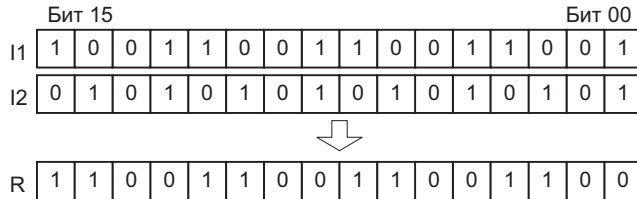


Контроллер	Номер инструкции	Операнды		
		I1	I2	R
	№г	первое слово	второе слово	результат
C200H-CPU-E	36	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H-CPU-ZE	036	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Описание**

Когда условие исполнения = 0, XORW не выполняется. Когда условие исполнения = 1, XORW производит операцию ИСКЛЮЧАЮЩЕЕ ИЛИ с содержимым I1 и I2 побитно и выдает результат в R.

**Пример**

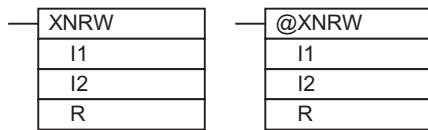


**Флаги**

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- EQ:** 1, когда результат равен нулю.
- N:** 1, когда бит 15 слова R = 1.

5.23.5 XNRW - Исключающее ИЛИ-НЕ

**Обозначение на ПКС**

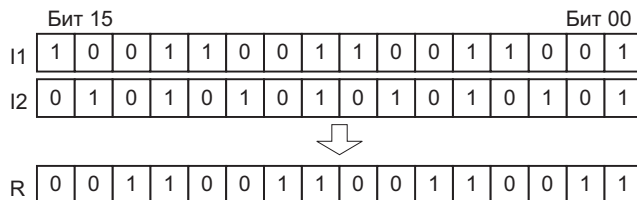


Контроллер	Номер инструкции	Операнды		
		I1	I2	R
	№г	первое слово	второе слово	результат
C200H-CPU-E	37	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H-CPU-ZE	037	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Описание**

Когда условие исполнения = 0, XNRW не выполняется. Когда условие исполнения = 1, XNRW производит операцию ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ с содержимым I1 и I2 побитно и выдает результат в R.

**Пример**



**Флаги**

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- EQ:** 1, когда результат равен нулю.
- N:** 1, когда бит 15 слова R = 1.

### 5.24 Управление подпрограммами и прерываниями

#### 5.24.1 Подпрограммы

Подпрограммы разбивают большие задачи управления на небольшие и позволяют повторно использовать набор блоки команд. Когда главная программа вызывает подпрограмму, управление передается к подпрограмме, и выполняются команды подпрограммы. Команды внутри подпрограммы пишутся аналогично главной программе. Когда исполнятся все подпрограммы, управление возвращается к главной программе в точку сразу за точкой, из которой была вызвана подпрограмма. (Если в подпрограмме не назначено что-либо другое).

Подпрограммы могут быть вызваны из прерываний или командой MCRO.

#### Прерывания

Подобно вызовам подпрограммы, прерывания прерывают обработку главной программы, и после завершения подпрограммы исполнение возвращается в данную точку. Прерывание вызывается либо из внешнего источника, такого как входной сигнал с блока входных прерываний, либо из прерывания по расписанию. В случае прерываний по расписанию сигнал прерывания повторяется через равные интервалы времени.

Хотя вызовы подпрограммы управляются из главной программы, подпрограммы, вызванные прерыванием, начинают обрабатываться после получения сигнала прерывания. В случае прерываний по расписанию интервал между прерываниями задается пользователем и не зависит от времени цикла ПК. Это свойство полезно для периодического контроля или выполнения исполнительской программы.

INT служит для управления сигналами прерываний с блока входных прерываний, и также для регулирования расписания при прерываниях по расписанию. INT предоставляет такие функции, как маскирование прерываний (чтобы они были записаны, но проигнорированы) и очистки прерываний.

Подробности о прерываниях см. 5.23.2.

#### MCRO

Команда MCRO позволяет одной подпрограмме (образцу) заменить несколько подпрограмм, имеющих идентичную структуру, но разные операнды. Поскольку несколько одинаковых программных секций могут управляться одной подпрограммой, количество шагов программы можно сильно сократить. Подробности о данной инструкции см. 5.23.5.

#### 5.24.2 Прерывания

C200HX/HG/HE поддерживает и входные прерывания, и прерывания по расписанию. Прерывания останавливают исполнение программы для вызова подпрограммы, которая либо должна исполняться немедленно (входные прерывания с блока входных прерываний) или периодически (прерывания по расписанию).

Есть 2 режима прерываний. В нормальном режиме ЦПУ ожидает, пока текущий процесс завершится перед остановом главной программы. В высокоскоростном режиме ЦПУ прерывает текущий процесс. Нормальный режим - это режим по умолчанию C200HX/HG/HE, но установочным параметром ПК можно выбрать высокоскоростной режим.

#### Нормальный режим прерываний

Следующие параметры используются для нормального режима прерываний.

DM 6620 

0	0	0	0
---	---	---	---

В нормальном режиме прерываний будет завершена обработка следующих функций, даже в случае прерываний. Прерывания будут обработаны, как только завершится текущий процесс.

- Обслуживание HOST LINK
- Обслуживание удаленных входов/выходов

- Обслуживание специальных входов/выходов
- Выполнение индивидуальных команд

Используйте этот режим при подпрограммах прерываний C200H без модификаций или, если возможно, с учетом времени реакции, требуемого для прерывания.

**Замечание** По умолчанию C200HX/HG/HE установлен в режим нормальных прерываний.

### Режим высокоскоростных прерываний

Следующие установочные параметры используются для режима высокоскоростных прерываний.

DM 6620 

0	0	0	1
---	---	---	---

Режим высокоскоростных прерываний

В режиме высокоскоростных прерываний обработка следующих функций будет прервана и программа прерываний будет обработана, как только вызовется прерывание.

- Обслуживание HOST LINK
- Обслуживание удаленных входов/выходов
- Обслуживание специальных входов/выходов
- Выполнение индивидуальных команд

Используйте этот режим, когда время реакции на прерывание должен быть точнее 1.0 мс.

При использовании режима высокоскоростных прерываний данные не обязательно будут корректными, поскольку обслуживание HOST LINK, обслуживание удаленных входов/выходов, обслуживание специальных входов/выходов и выполнение индивидуальных команд необязательно будет завершено, когда начнется отработка прерывания. Программа должна быть разработана так, чтобы позволить этого, если требуется. (Дальнейшие подробности см. главу о соответствии данных).

### Входные прерывания

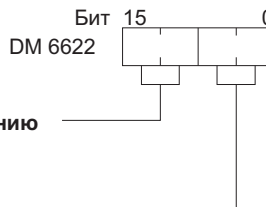
Входные прерывания выполняются, когда внешние входы получены с блока входных прерываний. На раме ЦПУ можно смонтировать до 2 блоков входных прерываний, и каждый блок входных прерываний дает 8 входов с номерами IN 0.. IN 7.

Входы IN0.. IN7 первого блока генерируют прерывания #00.. #07, а входы IN0.. IN7 второго блока генерируют прерывания #08.. #15. Другими словами, подпрограммы #00.. #15 исполняются, когда вызываются прерывания #00.. #15.

### Прерывания по расписанию

Прерывания по расписанию можно исполнять с интервалами, заданными с дискретностью 10 мс или 1 мс. Когда вызывается прерывание #99, исполняется подпрограмма #99.

Дискрета задания интервала прерывания задается в параметре DM 6622.



#### Разрешение задания параметров прерывания по расписанию

- 00: запретить (фиксированно на 10 мс)
- 01: Разрешить (задание в битах 00..07)

#### Задание параметров прерывания по расписанию

- 00: 10 мс
- 01: 1 мс

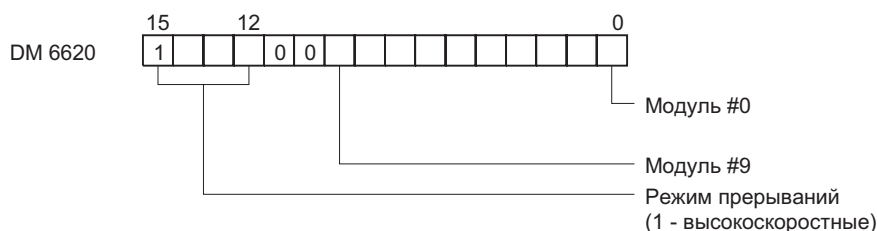
### Приоритет прерываний

Заданная подпрограмма будет исполнена при вызове прерывания. При генерации последующих прерываний во время исполнения подпрограммы прерывания, они будут обрабатываться после выполнения текущей подпрограммы. Если генерируется более одного прерывания (или ожидает своей очереди), соответствующие подпрограммы будут исполняться в порядке приоритетов.

Входное прерывание 1 Входное прерывание 2.. Входное прерывание 7 Прерывание по расписанию.

### Специальные входы/выходы в подпрограммах прерываний

Блоки специальных входов/выходов можно обновлять в подпрограммах прерываний командой IOFR. При использовании высокоскоростных прерываний обновление в нормальном цикле (обновление END и IEFR в главной программе) должно быть запрещено для блоков специальных входов/выходов, чтобы обновить их в подпрограмме прерываний. Признак ошибки программирования прерывания (FAL ошибка 8B) появляется, когда один и тот же блок специальных входов/выходов обновляется и в программе прерываний и в нормальном цикле, и тогда блок специальных входов/выходов не будут обновляться в подпрограмме прерываний. В установочных параметрах ПК содержится задание DM 6620, которые запрещают обновление в нормальном цикле для конкретных блоков специальных входов/выходов. Значения показаны на рисунке.



**Замечание** Запрещение обновления специальных входов/выходов в нормальном цикле для того, чтобы обновлять их в подпрограмме прерываний, необходимо только в высокоскоростном режиме. Запрещение обновления специальных входов/выходов в нормальном цикле при нормальном режиме прерываний можно игнорировать и блоки специальных входов/выходов будут обновляться и в нормальном цикле и в подпрограмме прерываний.

Время исполнения подпрограммы прерываний должно быть менее 10 мс в высокоскоростном режиме и использования блоков специальных входов/выходов, блоков HOST LINK или блоков удаленных входов/выходов. Признак ошибки программирования прерывания (FAL ошибка 8B) появляется, если время исполнения 10 мс или больше.

Время исполнения подпрограммы прерываний с самым большим временем прерывания выдается в SR 262 и номер подпрограммы с наибольшим временем исполнения выдается в SR 263.

Пример: 12.3 мс для подпрограммы #80

SR 262 

0	1	2	3
---	---	---	---

 Время исполнения подпрограммы прерывания с наибольшим временем исполнения

SR 263 

8	0	*	*
---	---	---	---

 Номер подпрограммы с наименьшим временем исполнения

**Замечание** Вышеупомянутый предел 10 мс не применим, когда используется нормальный режим прерываний или когда эти блоки не установлены.

### Корректность данных

Хотя корректность данных не является проблемой при исполнении нормальных операций арифметики или сравнения, она может стать проблемой при исполнении длинных команд, которые обрабатывают несколько слов, такие, как команда передачи блока, когда применяется режим высокоскоростных прерываний и одни и те же данные используются в главной программе и подпрограмме.

Данные могут быть некорректными в двух ситуациях:

Если операция записи данных в главной программе прерывается, и те же данные читаются в подпрограмме прерываний

Если операция чтения данных в главной программе прерывается, и те же данные читаются в подпрограмме прерываний.

Если Вы должны обрабатывать одни и те же данные и в главной программе, и в подпрограмме прерывания, используйте прием программирования, показанный на рисунке, для того, чтобы обеспечить сохранение корректности данных, т.е. маскирование прерывания при чтении/записи данных, которые также используются в подпрограмме прерывания.



Корректность данных может также стать проблемой, если прерывания происходят во время передачи данных при обслуживании блоков специальных входов/выходов, удаленных входов/выходов или систем HOST LINK. Во всех этих случаях данные могут быть некорректными.

Для сохранения корректности данных используйте следующие методы. Второй метод применим только для блоков специальных входов/выходов.

- Маскирование прерываний в главной программе при процессе перемещения данных в/из блоков и используйте эти измененные значения в программе прерываний.
- Используйте команду IORF в подпрограмме прерываний для обновления требуемых входов/выходов с блоков специальных входов/выходов и маскирование прерываний в главной программе при чтении/записи блоков специальных входов/выходов.

### 5.24.3 SBS - Вызов подпрограммы

#### Обозначение на ПКС



	Номер инструкции	Операнды		
Контроллер	Nr	N		
		номер подпрограммы		
C200H□-CPU□□-E	91	# (00..255)		
C200H□-CPU□□-ZE	091	# (00.255)		

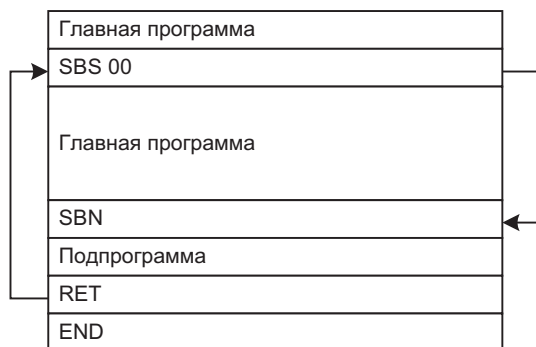
#### Ограничения

Номера подпрограмм 00..15 используются с подпрограммами прерываний, а подпрограмма 99 служит для прерываний по расписанию.

#### Описание

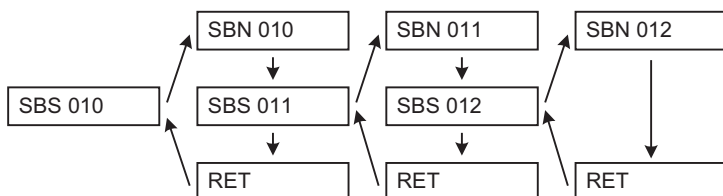
Подпрограмма выполняется путем помещения команды SBS в главной программе в точке, в которой необходим вызов подпрограммы. Номер, который задается вместе с SBS, указывает на номер требуемой подпрограммы. Когда выполняется команда SBS (т.е. условие исполнения = 1), то выполняются команды между SBN с таким же самым номером и первой командой RET после нее, прежде чем исполнение программы возвратится к команде, следующей за SBS, сделавшей вызов.



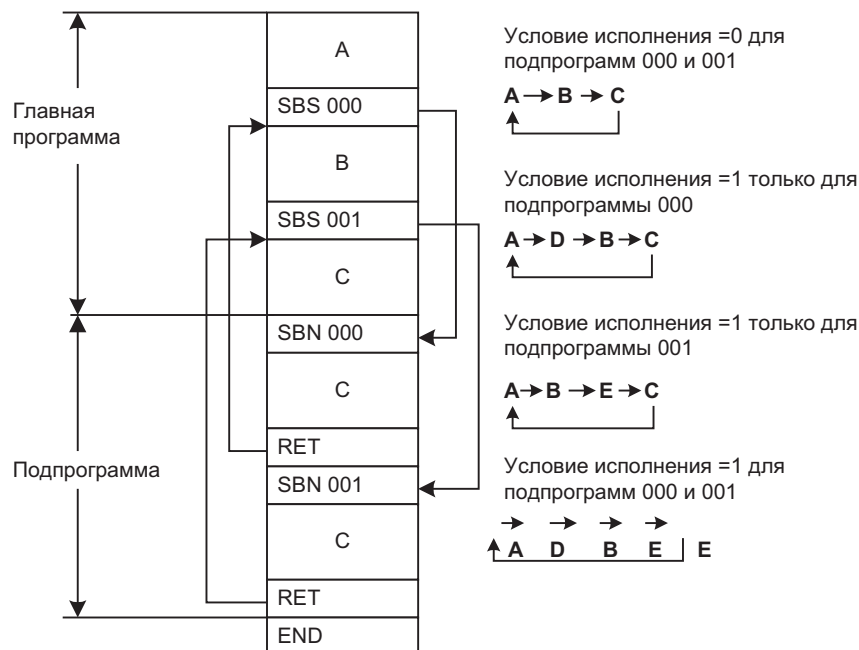


SBS можно использовать в программе столько раз, сколько требуется, т.е. одну и ту же подпрограмму можно вызвать из разных мест программы.

SBS можно также поместить в подпрограмму для вызова одной подпрограммы из другой, т.е. подпрограммы могут быть вложенными. После завершения второй подпрограммы (т.е. выполнялась команда RET), выполнение программы возвращается к исходной подпрограмме, которая выполняется и возвращается в главную программу. Возможны 16 уровней вложений. Подпрограмма не может вызывать саму себя. (т.е. нельзя запрограммировать SBS 00 внутри подпрограммы, определенной командой SBN 00). Следующая диаграмма иллюстрирует два уровня вложения.



Следующая диаграмма иллюстрирует процесс исполнения программы для различных условий исполнения для двух SBS.



**Замечание** Признак нефатальной ошибки (код 8В) будет генерироваться, если время исполнения подпрограммы превысит 10 мс.

**Флаги**

- ER:** Не существует подпрограммы с заданным номером.
- Подпрограмма вызывает сама себя
- Вызвана активная подпрограмма.

**Внимание!** SBS не будет выполняться и подпрограмму нельзя вызвать, если флаг ER = 1.

#### 5.24.4 SBN и RET - Начало подпрограммы и возврат из подпрограммы

##### Обозначение на ПКС

— SBN N

	Номер инструкции	Операнды
Контроллер	№	N номер подпрограммы
C200H-CPU-E	92	# (00..255)
C200H-CPU-ZE	092	# (00..255)

##### Обозначение на ПКС

— RET

	Номер инструкции	Операнды
Контроллер	№	
C200H-CPU-E	93	
C200H-CPU-ZE	093	

##### Ограничения

Каждый номер можно использовать в SBN только раз.

##### Описание

SBN служит для обозначения начала подпрограммы. RET служит для обозначения конца подпрограммы. Каждая подпрограмма идентифицируется номером подпрограммы N, который программируется как определитель SBN. Такой же номер используется командой SBS, которая вызывает данную подпрограмму (см. 5-23-3). RET не требует номера.

Все подпрограммы должны программироваться в конце главной программы. Когда запрограммированы одна или более подпрограмм, главная программа будет исполняться до первой команды SBS и затем возвратится на адрес 00000 для следующего цикла. Если подпрограммы на вызываются командой SBS, они не выполняются.

END должен ставиться в конце последней подпрограммы, т.е. после последнего RET. В других местах программы данная команда не требуется.

##### Предосторожности

Если SBN ошибочно помещена в главной программе, она запретит исполнение после своего адреса, т.е. выполнение программы возвратится к началу, если встретилась команда SBN.

Если DIFU или DIFD стоят в подпрограмме, битовый операнд установится в состояние 0 только в следующий раз исполнения подпрограммы, т.е. битовый операнд может оставаться в состоянии 1 больше, чем 1 цикл.

##### Флаги

Данные команды не оказывают прямого действия на флаги.

#### 5.24.5 MCRO - Функция Макро

##### Обозначение на ПКС

— <span style="border: 1px solid black; padding: 2px;">MCRO</span>	— <span style="border: 1px solid black; padding: 2px;">@MCRO</span>
N	N
I1	I1
O1	O1

## 5.24 Управление подпрограммами и прерываниями

Контроллер	Номер инструкции	Операнды		
		N	I1	O1
	№г	номер подпрограммы	первое слово входных данных	первое слово выходных данных
C200H□-CPU□□-E	99	# (00..255)	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	099	# (00..255)	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

### Ограничения

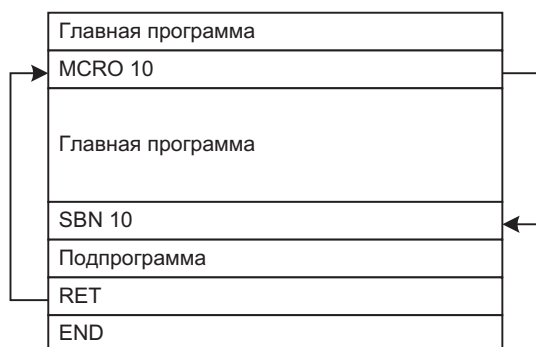
I1.. I1+3 должны лежать в одной области данных (как и O1.. O1+3).

### Описание

Команда MCRO позволяет написать одну подпрограмму вместо нескольких подпрограмм, которые имеют одинаковую структуру, но разные операнды. Есть 4 слова входа SR 290.. SR 293 и 4 слова выхода SR 294.. SR 297, которые отведены для MCRO. Данные 8 слов используются в подпрограмме и берут свое содержимое из I1.. I1+3 и O1.. O1+3 при исполнении подпрограммы.

Когда условие исполнения = 0, MCRO не выполняется. Когда условие исполнения = 1, MCRO копирует содержание I1.. I1+3 в SR 290.. SR 293, копирует содержание O1.. O1+3 в SR 294.. SR 297 и затем вызывает и исполняет подпрограмму, заданную в N. По завершении подпрограммы содержимое SR 294.. SR 297 передается обратно в O1.. O1+3 перед завершением подпрограммы.

В следующем примере содержимое DM 0010.. DM 0013 копируется в SR 290.. SR 293, содержимое DM 0020.. DM 0023 копируется в SR 294.. SR 297 и вызывается на исполнение подпрограмма 10. После завершения подпрограммы содержимое SR 294.. SR 297 передается обратно в DM 0020.. DM 0023.



- Замечание**
1. Подпрограммы для MCRO программируются, как и другие подпрограммы, за исключением того, что содержимое SR 290.. SR 297 передается из указанных входов и на указанные выходы.
  2. Не только внешние слова входов/выходов, но и внутренние слова можно использовать для I1 и O1.
  3. SR 290.. SR 297 можно использовать, как рабочие биты, если команда MCRO не используется.
  4. Команды MCRO можно вкладывать друг в друга, но при этом нужно быть внимательным при сохранении данных входов/выходов, поскольку команды используются одни и те же 8 слов входа/выхода (SR 290.. SR 297).

### Предосторожности

MCRO можно использовать только для секций программы, которые можно записать с использованием 4 и меньше последовательных входных слов и/или 4 и менее последовательных выходных слов. Так что необходимо принимать во внимание систему и программу, чтобы полностью использовать преимущества команды MACRO.

Будьте внимательны, чтобы слова входов и выходов точно соответствовали критериям слов входа и выхода.

### Флаги

- ER:** Заданный номер подпрограммы не существует.  
Операнд превысил границу области данных.

## 5.24 Управление подпрограммами и прерываниями

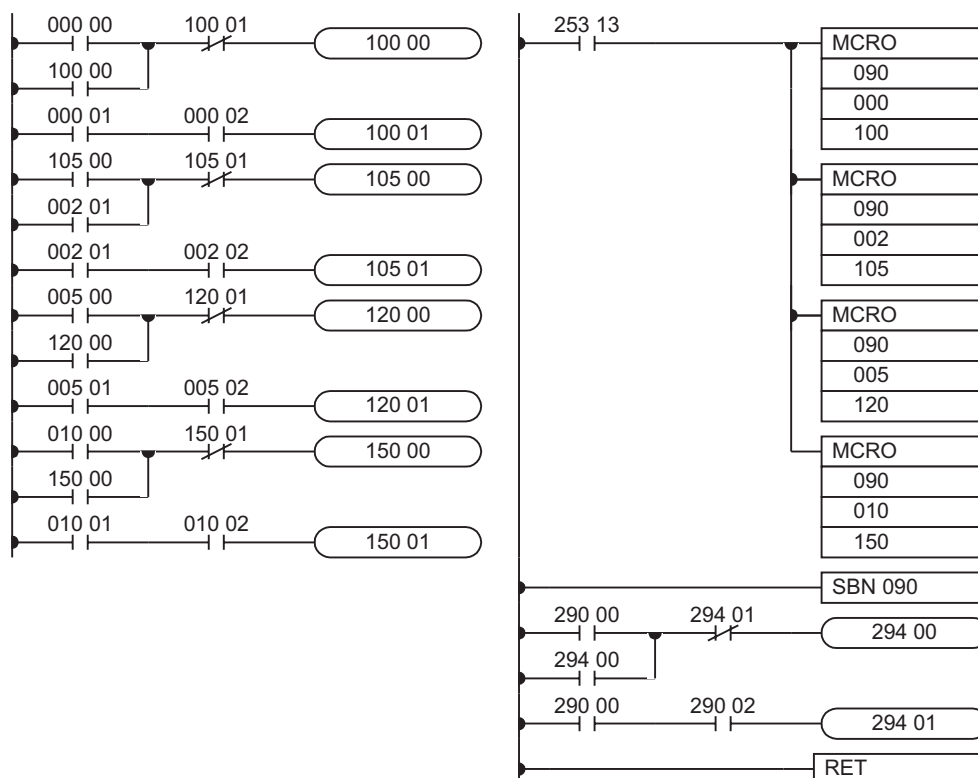
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

Подпрограмма вызывает саму себя.

Вызвана работающая подпрограмма.

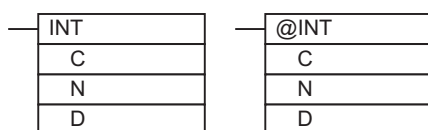
### Пример

В следующем примере показано использование 4 команд MCRO, которые работают с одной и той же подпрограммой. Секция программы слева показывает ту же программу без использования команды MCRO.



### 5.24.6 INT - Управление прерываниями

#### Обозначение на ПКС



Контроллер	Номер инструкции	Операнды		
		C	N	D
		слово состояния	тип прерывания	контрольное слово
C200□□-CPU□□-E	89	# (000, 001, 002, 100, 200)	# (000, 001, 004)	IR, SR, HR, AR, LR, TC, DM, #
C200□□-CPU□□-ZE	089	# (000, 001, 002, 100, 200)	# (000, 001, 004)	IR, SR, HR, AR, LR, TC, DM, EM, #

#### Ограничения

D должно быть в диапазоне #0000.. #00FF, когда N = 000 и C = 000 или 001

D должно быть двоично-десятичным числом в диапазоне #0001.. #9999, когда N = 004 и C = 000 или 001.

#### Описание

INT служит для управления прерываниями и осуществляет одну из 11 функций, зависящих от значений C и N. Как показано в следующих таблицах, 6 функций действуют на входные прерывания, три на прерывания по расписанию и остальные два для маскирования и размаскирования всех прерываний.

## 5.24 Управление подпрограммами и прерываниями

Прерывание	С	Функция INT	Комментарии
Входные прерывания от блока входных прерываний 0 (N=000)	000	Маскировать/размаскировать входные прерывания	Биты 00.. 07 слова D указывают входы 00.. 07
	001	Очистить входные прерывания	
	002	Читать текущее состояние маски	Состояние записано в D
Входные прерывания от блока входных прерываний 1 (N=001)	000	Маскировать/размаскировать входные прерывания	Биты 00.. 07 слова D указывают входы 00..07
	001	Очистить входные прерывания	
	002	Читать текущее состояние маски	Состояние записано в D
По расписанию (N=004)	000	Задать интервал прерываний	-
	001	Задать время первого прерывания	-
	002	Читать интервал прерываний	-

Следующие 2 функции зависят только от значения С.

Значение	Функция INT
100	Замаскировать все прерывания
200	Размаскировать все прерывания

### Маскировать/размаскировать входные прерывания (N=000 или 001, С=000)

Задайте N=000 для блока входных прерываний 0 и N=001 для блока входных прерываний 1. Данная функция служит для маскирования и размаскирования входных прерываний 00.. 07. Замаскированные прерывания запоминаются, но игнорируются. Когда один из входов замаскирован, программа прерываний для него будет исполняться только после размаскирования прерываний (если только он предварительно не очищен командой INT с С = 001 и N=000).

Установите соответствующий бит в D в 0 для размаскирования или 1 для маскирования входа прерывания. Биты 00.. 07 соответствуют 00.. 07.

### Очистить входные прерывания (N=000 или 001, С=001)

Задайте N=000 для блока входных прерываний 0 и N=001 для блока входных прерываний 1. Данная функция служит для очистки входов входных прерываний 00.. 07. Поскольку входы с запросами прерываний запомнены, если не очистить память о запросах входных прерываний, замаскированные прерывания будут обслужены после снятия маски.

Установите соответствующий бит в D в 1 для очистки входного прерывания. Биты 00.. 07 соответствуют 00.. 07.

### Читать текущее состояние маски (N=000 или 001, С=002)

Задайте N=000 для блока входных прерываний 0 и N=001 для блока входных прерываний 1. Данная функция служит для записи текущего состояния маски входов прерываний 00.. 07. в слово D. Если вход замаскирован, соответствующий бит будет 1. (Биты 00.. 07 соответствуют 00.. 07.).

### Задать интервал прерываний (N=004, С=000)

Данная функция служит для задания интервала между прерываниями по расписанию. Содержимое D (двоично-десятичное число: 0001.. 9999) умножается на дискрету времени прерывания по расписанию (1 мс или 10 мс) для получения интервала.

Дискрета времени прерывания по расписанию задается в DM 6622. Подробности о задании дискреты см. 3.6.4.

### Задать время первого прерывания (N=004, С=001)

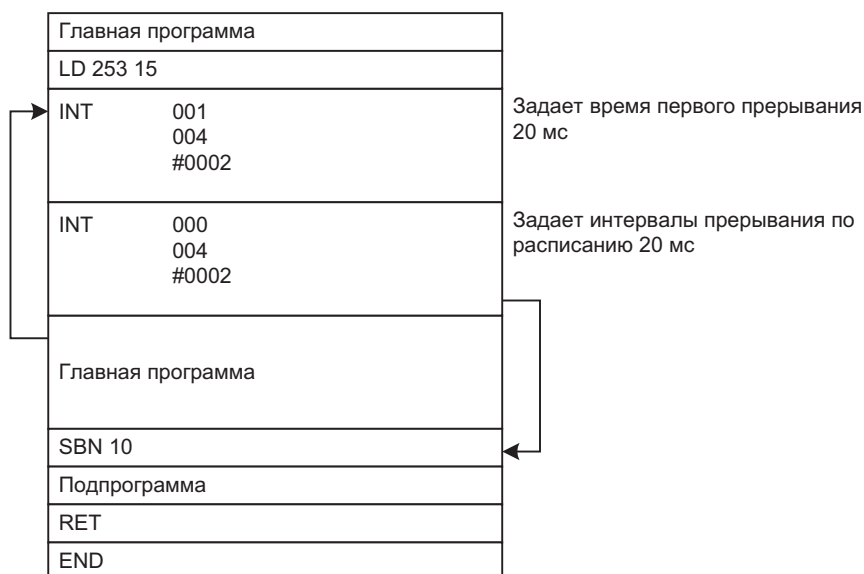
Данная функция служит для задания времени первого прерывания по расписанию. Содержимое D (двоично-десятичное число: 0001.. 9999) умножается на дискрету времени прерывания по расписанию (1 мс или 10 мс) для получения времени первого прерывания по расписанию.



### Пример 2: прерывание по расписанию

В данном примере показано, как задать интервал прерываний по расписанию. Подпрограммы прерываний будут выполняться с фиксированными интервалами, независимо от размещения в цикле ЦПУ. Данное прерывание полезно для таких программных секций, как программы регулярного контроля.

Прерывание по расписанию запрещено при пуске программы (интервал прерываний по расписанию =0), так что время первого прерывания и интервал нужно задать командой INT при N=004 и C=001/000. На следующей диаграмме подпрограмма будет выполняться каждые 20 мс, если дискрета времени прерывания по расписанию задана 10 мс в DM 6622.



**Замечание** В зависимости от параметра DM 6621 обслуживание HOST LINK, удаленных входов/выходов, специальных входов/выходов и индивидуальных команд будет завершено перед исполнением подпрограммы. Подробности см. 5.23.1.

## 5.25 Команды секции STEP

Команды секции STEP: STEP и SNXT используются совместно для задания точек разрыва между секциями в больших программах, чтобы секции можно было выполнять как блоки и сбрасывать после исполнения. Секция программы обычно определяется для соответствия физическому процессу. (См. пример применения в данной главе.). Команды секции STEP аналогичны остальным, за исключением того, что некоторые команды (например, IL/ILC, JMP/JME) нельзя включать между ними.

### 5.25.1 STEP и SNXT - Задание секции STEP / Запуск секции STEP

#### Обозначение на ПКС



Номер инструкции		Операнды		
Контроллер	№	B	контрольный бит	
C200H□-CPU□□-E	08	IR, SR, HR, AR, LR		
C200H□-CPU□□-ZE	008	IR, SR, HR, AR, LR		

#### Обозначение на ПКС



Номер инструкции		Операнды		
Контроллер	№	B	контрольный бит	
C200H□-CPU□□-E	09	IR, SR, HR, AR, LR		
C200H□-CPU□□-ZE	009	IR, SR, HR, AR, LR		

#### Ограничения

Все биты управления должны быть в одном и том же слове и должны располагаться последовательно.

IR 29800..29915 нельзя использовать для DB

#### Описание

STEP использует бит управления в областях IR и HR для определения начала секции программы, называемой STEP. Команда STEP не требует условий исполнения, т.е. ее исполнение управляется битом управления. Для пуска исполнения секции STEP служит SNXT с тем же самым управляющим битом, что и в STEP. Если условие исполнения для SNXT = 1, выполняется секция STEP с тем же самым битом управления. Если условие исполнения = 0, секция STEP не выполняется. Команда SNXT должна быть записана в программе таким образом, чтобы она исполнялась до того, как программа достигнет секции STEP, которую она вызывает. Она используется в различных местах перед секцией STEP для управления секцией в соответствии с двумя различными условиями (см. пример 2 ниже). Никакая секция STEP программы, не запущенная командой SNXT, не выполняется.

Когда в программе используется SNXT, исполнение секции STEP будет продолжаться, пока STEP не выполнится без бита управления. Перед STEP без управляющего бита должна стоять SNXT с фиктивным управляющим битом. Фиктивным управляющим битом может служить любой неиспользуемый бит IR или HR. Им не должен быть бит, используемый STEP.

Исполнение секции STEP завершается либо исполнением следующей команды SNXT, либо установкой в 0 управляющего бита данной секции STEP (см. пример 3 далее). Когда секция STEP выполнена, все биты IR и HR устанавливаются в 0 и все таймеры секции сбрасываются в свои заданные значения. Счетчики, регистры сдвига и биты, используемые в KEEP сохраняют свое состояние. В примере показаны две простых секции STEP.





Адрес	Инструкция	Операнд
00000	LD	000 00
00001	SNXT	LR 2000
00002	STEP	LR 2000
Секция STEP, управляемая LR 2000		
00100	LD	000 01
00101	SNXT	LR 2001
00102	STEP	LR 2001
Секция STEP, управляемая LR 2001		
00200	LD	000 02
00201	SNXT	LR 2002
00202	STEP	

Секции STEP можно программировать последовательно. Каждая секция должна начинаться с STEP и обычно заканчиваться SNXT (см. далее пример 3, исключение). Когда секции программируются последовательно, возможны 3 режима исполнения: последовательный, с ветвлением или параллельный. Условия исполнения и позиция для SNXT определяют, как будут исполняться секции. Три примера иллюстрируют три режима исполнения секций.

### Предосторожности

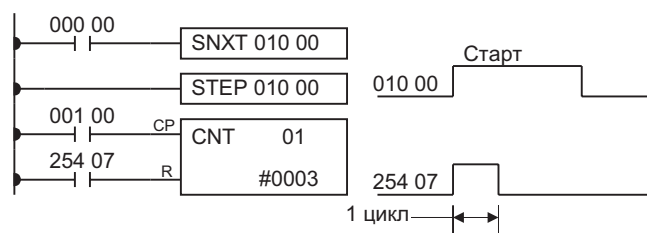
Команды INTERLOCK, переходы, SBN, END нельзя использовать внутри секций STEP.

Биты, служащие управляющими битами, нельзя использовать больше нигде в программе, (если только они не используются для управления секцией STEP) (см. пример 3). Все биты управления должны находиться в одном слове и располагаться последовательно.

Если в качестве управляющих битов используются биты IR или LR, их состояние будет потеряно при отключении питания. Если необходимо сохранить состояние для возобновления работы с той самой секции STEP, нужно использовать биты HR.

### Флаги

**25407:** Флаг пуска секции STEP; установлен в 1 на 1 цикл при исполнении STEP и служит в случае необходимости для сброса счетчиков в секции STEP, как показано на следующем примере.



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	SNXT	010 00
00002	STEP	010 00
00003	LD	010 00
00004	LD	254 07
00005	CNT	01
		#0003

**Примеры**

Следующие три примера показывают три возможных типа управления, использующих команды STEP.

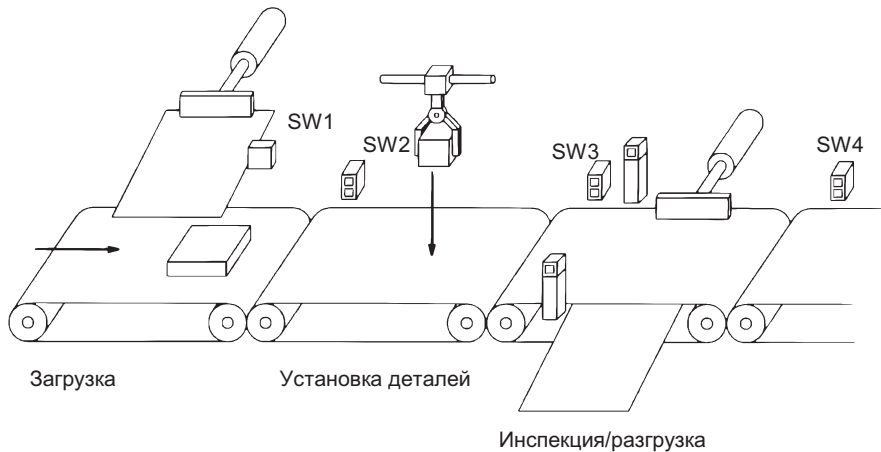
Пример 1 показывает последовательное исполнение;

Пример 2 показывает исполнение ветвления;

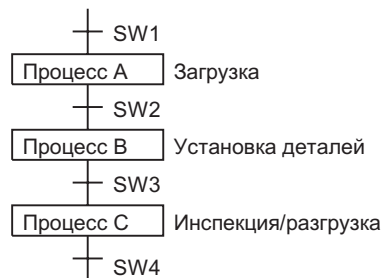
Пример 3 показывает параллельное исполнение;

**Пример 1: последовательное исполнение**

Следующий процесс требует, чтобы три процесса, загрузка, установка деталей и инспекция/разгрузка выполнялась последовательно, причем каждый процесс сбрасывался перед исполнением следующего. Датчики (SW1, SW2, SW3 и SW4) установлены, чтобы сигнализировать, когда процессы должны начинаться и заканчиваться.



На диаграмме показан алгоритм работы и выключатели, которые управляют процессом.



Программа для этого процесса, показанная ниже, использует основной тип программирования STEP: каждый STEP завершается уникальным SNTX, который начинает следующий STEP. Каждый STEP начинается, когда переключатель, который указывает, что предыдущий STEP окончен, = 1.

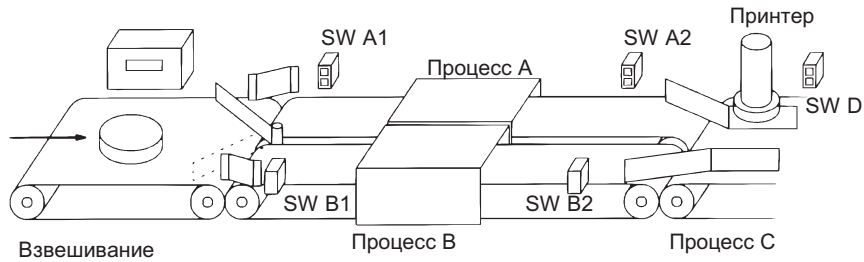


SW1: 000 01  
 SW2: 000 02  
 SW3: 000 03  
 SW4: 000 04

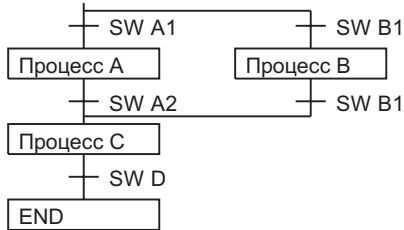
Адрес	Инструкция	Операнд
00000	LD	000 01
00001	AND NOT	000 02
00002	SNXT	HR 0000
00003	LD NOT	000 01
00004	AND	000 02
00005	SNXT	HR 0001
00006	STEP	HR 0000
Процесс А		
00100	LD	000 03
00101	SNXT	HR 0002
00102	STEP	HR 0001
Процесс В		
00150	LD	000 04
00151	SNXT	HR 0002
00152	STEP	HR 0002
Процесс С		
00200	LD	000 05
00201	SNXT	HR 0003
00202	STEP	

**Пример 2: исполнение процесса с ветвлением**

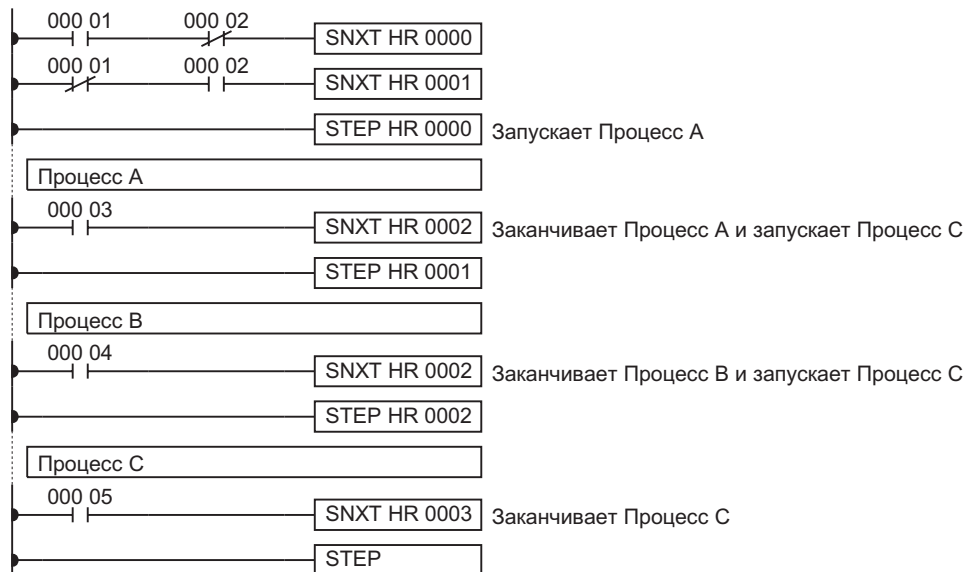
Следующий процесс требует, чтобы перед печатью деталь обрабатывалась одним из двух способов в зависимости от веса. Процесс печати одинаков независимо от первого процесса. Датчики расположены таким образом, чтобы сигнализировать о начале и завершении процесса.



На диаграмме показан алгоритм работы и выключатели, которые используются для управления процессом. Здесь, используется либо процесс А, либо процесс В в зависимости от состояния SW A1 и SW B1.



Программа для этого процесса, показанная ниже, начинается двумя командами SNXT, которые начинают процессы А и В. Поскольку из-за способа программирования 00001 (SW A1) и 00002 (SW B1) будет выполняться только одна команда из двух для запуска либо процесса А, либо В. Оба STEP оканчиваются командой SNXT, которая начинает процесс С.



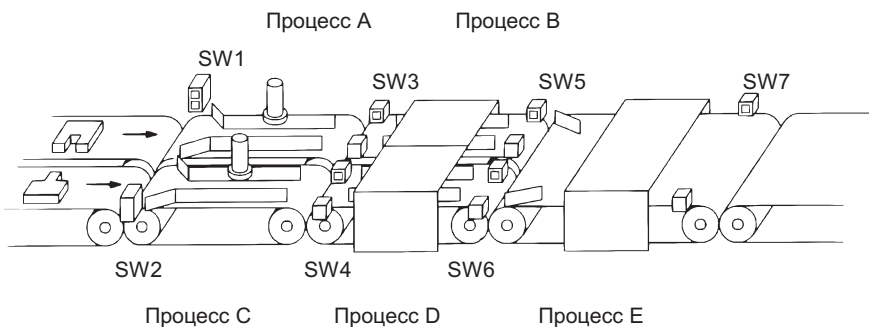
SW A1: 000 01  
 SW A2: 000 03  
 SW B1: 000 02  
 SW B2: 000 04  
 SW D: 000 05

Адрес	Инструкция	Операнд
00000	LD	000 01
00001	AND NOT	000 02
00002	SNXT	HR 0000
00003	LD NOT	000 01
00004	AND	000 02
00005	SNXT	HR 0001
00006	STEP	HR 0000

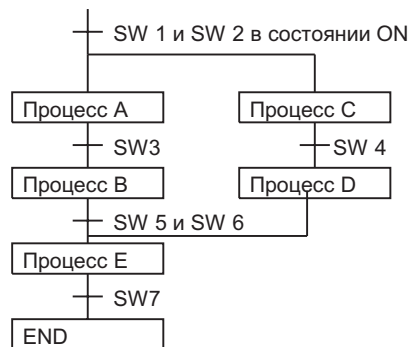
Адрес	Инструкция	Операнд
Процесс А		
00100	LD	000 03
00101	SNXT	HR 0002
00102	STEP	HR 0001
Процесс В		
00150	LD	000 04
00151	SNXT	HR 0002
00152	STEP	HR 0002
Процесс С		
00200	LD	000 05
00201	SNXT	HR 0003
00202	STEP	

**Пример 3: параллельное исполнение**

Следующий процесс требует, чтобы два изделия одновременно проходили обработку, перед тем, как они объединятся в пятом процессе. Датчики расположены таким образом, чтобы сигнализировать о начале и завершении процесса.

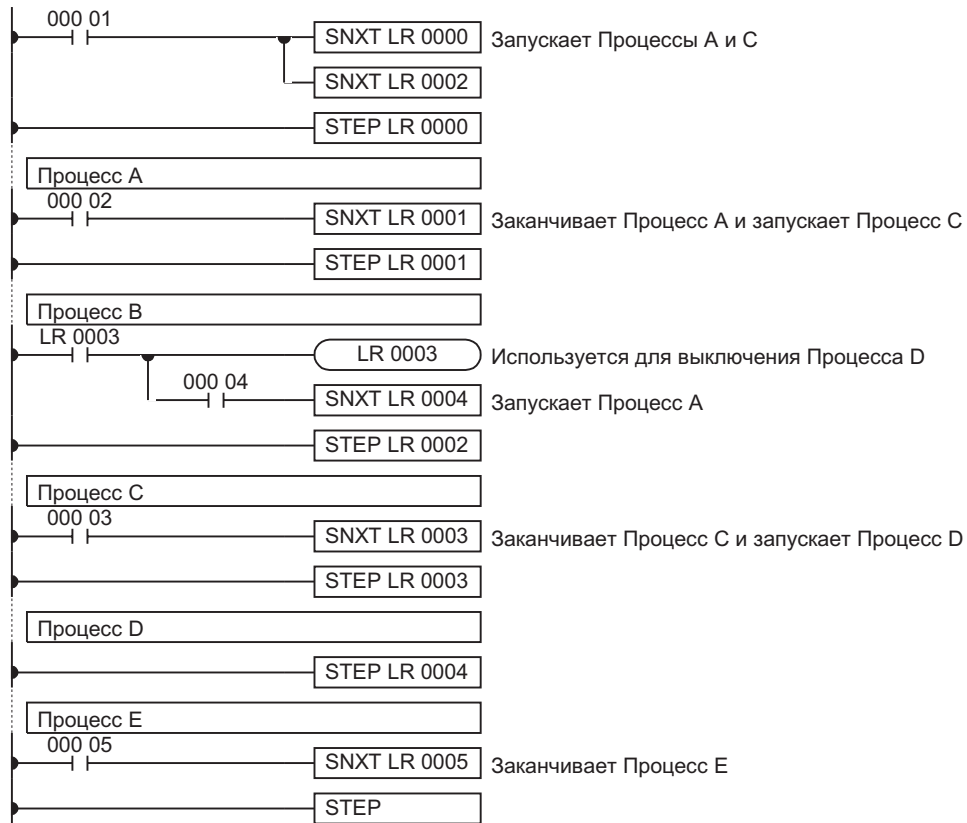


На диаграмме показан алгоритм работы и выключатели, которые используются для управления процессом. Здесь, процесс А и процесс С начинаются вместе. Когда оканчивается процесс А, начинается процесс В, Когда оканчивается процесс С, начинается процесс D. Когда оканчиваются оба процесса В и D, начинается процесс Е.



Программа для данной операции, показанная ниже, начинается двумя командами SNXT, которые запускают процессы А и С. Эти команды разветвляются от одной командной строки и всегда выполняются вместе, запуская секции STEP для А и С. Когда эти STEP окончены и для А и для С, сразу же запускаются процессы В и D.

Когда оба процесса В и D завершены, (т. е. когда состояние обоих из них = 1, но SW5 и SW 6 = 1), процессы В и D оба сбрасываются командой SNXT в конце программы процесса В. Хотя в конце процесса D нет SNXT, бит управления для него включается в 0 исполнением SNXT LR 0004. Это происходит из-за того, что OUT для LR 0003 сброшен в STEP командой SNXT LR 0004, т.е. LR 0003 включается в 0, когда выполняется SNXT LR 0004. Таким образом, процесс В сбрасывается прямо, а процесс D косвенно перед исполнением STEP для процесса Е.



SW1, SW2: 000 01  
 SW3: 000 02  
 SW4: 000 03  
 SW5, SW6: 000 04  
 SW7: 000 05

Адрес	Инструкция	Операнд
00000	LD	000 01
00001	SNXT	LR 0000
00002	SNXT	LR 0002
00003	STEP	LR 0000
Процесс А		
00100	LD	000 02
00101	SNXT	LR 0001
00102	STEP	LR 0001
Процесс В		
00200	LD	LR 0003
00201	OUT	LR 0003
00202	AND	000 04
00203	SNXT	LR 0004
00204	STEP	LR 0002
Процесс С		
00300	LD	000 03
00301	SNXT	LR 0003
00302	STEP	LR 0003
Процесс D		
00400	STEP	LR 0004

## 5.25 Команды секции STEP

---

Адрес	Инструкция	Операнд
Процесс E		
00500	LD	000 05
00501	SNXT	LR 0005
00502	STEP	

## 5.26 Специальные команды

Команды, описанные в данной главе, используются для различных операций, включая программирование сообщений пользователя, подсчет битов в состоянии 1, установка контрольного таймера и обновление входов/выходов во время исполнения программы.

### 5.26.1 FAL и FALS - Предупреждение о нефатальной ошибке и Предупреждение о фатальной ошибке

#### Обозначение на ПКС

—  —

Номер инструкции		Операнды		
Контроллер	№г	N		
		номер ошибки		
C200H□-CPU□□-E	06	# (00..99)		
C200H□-CPU□□-ZE	006	# (00..99)		

#### Обозначение на ПКС

—

Номер инструкции		Операнды		
Контроллер	№г	N		
		номер ошибки		
C200H□-CPU□□-E	07	# (00..99)		
C200H□-CPU□□-ZE	007	# (00..99)		

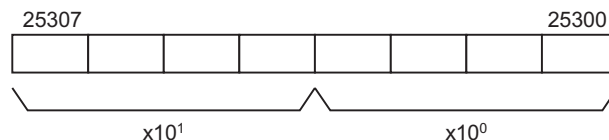
#### Ограничения

FAL и FALS используют одни и те же номера FAL. Следите, чтобы один номер использовался либо в FAL, либо в FALS, но не в обоих.

#### Описание

FAL и FALS используются для того, чтобы выдавать номера ошибок при работе, обслуживании и наладке. Когда условие исполнения для этих команд = 1, они выдают номер FAL в биты 00.. 07 SR 253. Номер может иметь значения 00.. 99 и является определителем команд FAL и FALS. FAL с определителем 00 используется для сброса данной области. (см. ниже).

#### Область ошибок FAL



FAL выдает признак нефатальной ошибки, FALS выдает признак фатальной ошибки. Когда выполняется FAL с условием 1, мигает индикатор ALARM/ERROR на передней панели ЦПУ, но операции ПК продолжаются. Когда выполняется FALS с условием 1, индикатор ALARM/ERROR на передней панели ЦПУ горит, и операции ПК прекращаются.

Система также генерирует коды ошибок в область FAL.

#### Сброс признаков ошибок

Все коды ошибок FAL будут храниться в памяти, хотя доступ в области FAL будет только к одной. Для доступа к другим кодам FAL нужно сбросить область FAL командой FAL 00. Каждый раз после выполнения FAL 00 код следующий ошибки будет посылаться в область FAL, стирая код той, которая была раньше. Коды ошибок FAL запоминаются и вызываются в следующем порядке:

первый сгенерированный код; самый меньший номер FAL, больший, чем первый код; самый меньший номер FAL, меньший, чем первый код.

#### Сброс сообщений

FAL 00 служит также для стирания сообщений, запрограммированных командой MSG.



Если область FAL нельзя очистить, что происходит после выполнения FALS, сначала устраните причину ошибки и далее очистите область FAL с помощью программатора (см. 4.6.5).

### 5.26.2 SCAN - Время цикла

#### Обозначение на ПКС

SCAN	@SCAN
Mi	Mi
000	000
000	000

Номер инструкции		Операнды		
Контроллер	№	Mi	000	000
		задание	не используется	не используется
C200H□-CPU□□-E	18	IR, SR, HR, AR, LR, TC, DM, #		
C200H□-CPU□□-ZE	018	IR, SR, HR, AR, LR, TC, DM, EM, #		

#### Ограничения

Mi должно быть двоично-десятичным числом. Используются только 3 правые (младшие) цифры Mi.

#### Описание

Команда SCAN служит для задания минимального времени цикла. Mi является минимальным временем цикла, задаваемое с дискретностью 0.1 мс, например, если Mi=1200, минимальное время цикла будет 120 мс. Возможный диапазон задания 000.0..999.9 мс.

Если текущее время цикла меньше, чем заданное командой SCAN, ЦПУ будет ожидать, пока истечет заданное время, прежде чем начать следующий цикл. Если текущее время цикла больше, чем заданное время, заданное время будет игнорироваться, и программа будет исполняться до завершения.

#### Флаги

**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

Mi не является двоично-десятичным числом.

### 5.26.3 TRSM - Выборка памяти трассировки

#### Обозначение на ПКС

TRSM
------

Номер инструкции		Операнды		
Контроллер	№			
C200H□-CPU□□-E	45			
C200H□-CPU□□-ZE	045			

Трассировка данных служит для облегчения отладки программ. Для установки и использования трассировки данных необходимо иметь управляющий компьютер с LSS. С программатора трассировка данных невозможна. Подробно трассировка данных описана в Инструкции по работе с LSS. В данной главе показан символ TRSM для релейно-контактной схемы и дан пример программы.

#### Описание

TRSM служит в программе для пометки мест, куда должны быть записаны специальные параметры в памяти трассировки. Для трассировки можно назначить до 12 битов и до 3 слов (Подробности см. в Инструкции по работе с LSS).

TRSM управляется не условиями исполнения, а двумя битами в области AR: AR 2515 и AR 2514. AR 2515 - бит начала выборки. Этот бит устанавливается в 1 для начала

процесса задания выборки для трассировки. Бит начала выборки не может устанавливаться в 1 из программы, а только с периферийного устройства.

AR 2515 - бит начала трассировки. Когда он установлен, назначенные данные загружаются в память трассировки. Бит начала трассировки можно устанавливать в 1 либо из программы, либо с периферийного устройства. Можно вводить положительное или отрицательное смещение для изменения фактической точки, откуда будет начинаться трассировка.

Данные можно записывать тремя путями. TRSM можно поместить в одном или нескольких местах программы для индикации, где нужно трассировать назначенные данные. Если TRSM не используется, назначенные данные будут трассироваться после исполнения END. Третий метод включает задание интервала таймера с периферийных устройств, так что назначенные данные будут трассироваться через регулярные интервалы времени независимо от времени цикла (см. Инструкцию по работе с).

TRSM можно встроить в любое место программы, либо число раз. Данные в памяти трассировки можно просматривать периферийным устройством, управляющим компьютером и т. д.

#### Биты и флаги управления области AR

Во время трассировки данных используются далее указанные биты и флаги управления. Флаг трассировки будет в состоянии 1 во время операции трассировки. Флаг завершения трассировки включится в 1, когда будет протрассировано достаточно данных для заполнения памяти трассировки.

Флаг	Функция
AR 2515	Бит начала выборки
AR 2514	Бит начала трассировки
AR 2513	Флаг ТРАССИРОВКА
AR 2512	Флаг завершения трассировки

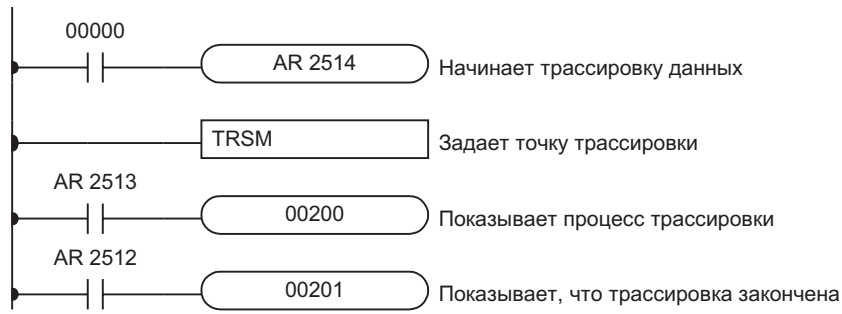
**Внимание!** TRSM не будет исполняться внутри блока перехода JMP - JME, если условия перехода = 0.

#### Пример

Следующий пример показывает базовую программу и операцию для трассировки данных. Принудительно включите бит начала выборки (AR 2515) для начала выборки. Данный бит должен быть установлен в 1 из программы. Данные читаются и загружаются в память трассировки.

Когда IR 00000 = 1, Бит начала трассировки (AR 2514) также устанавливается в 1, и ЦПУ ищет смещение и соответственно маркирует память трассировки. Это может значить, что некоторые уже готовые образцы будут записаны в память трассировки (отрицательное смещение), или будет сделано еще несколько выборок перед тем, как они будут записаны (положительное смещение).

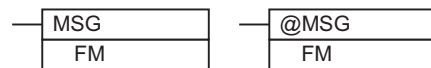
Эталонные данные записываются в память трассировки, переходя в начало области памяти сразу после достижения конца и продолжая до маркера старта. Это может значить, что ранее записанные данные (т.е. данные из образца, который встречается перед указателем старта) переписаны (это особенно верно, если смещение положительно). Отрицательное смещение не может быть таким, что требуемые данные выполнились перед началом выборки.



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	OUT	AR 2514
00002	TRSM	
00003	LD	AR 2513
00004	OUT	002 00
00005	LD	AR 2512
00006	OUT	002 01

### 5.26.4 MSG - Сообщение

#### Обозначение на ПКС



	Номер инструкции	Операнды
Контроллер	№	FM первое слово сообщения
C200H□-CPU□□-E	46	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	046	IR, SR, HR, AR, LR, TC, DM, EM

#### Ограничения

FM и FM+7 должны находиться в одной области памяти.

#### Описание

Когда условие исполнения = 0, MSG не выполняется. Когда условие исполнения = 1, MSG читает 8 слов расширенного кода ASCII FM.. FM+7 и индицирует сообщение на программаторе. Индицируемое сообщение может быть длиной до 16 знаков, т.е. каждый знак кода ASCII требует 8 бит (2 цифры). Коды ASCII см. Приложение H. В данный код включены также японские знаки катаканы.

Если длина сообщения меньше 8 слов, его можно прервать в любом месте, вводя "OD". Когда в сообщении встречается OD, слова больше не читаются, и ячейки памяти, которые нормально были бы использованы для сообщений, можно использовать для других целей.

#### Буферизация и приоритет сообщений.

В памяти можно забуферизовать до трех сообщений. После загрузки в буфер они индицируются на принципе "первый вошел, первый вышел". Поскольку возможно, что в одном цикле выполнилось более 3 команд MSG, основанная на область, в которой загружены сообщения, для выбора тех, которые требуется загрузить в буфер.

Приоритет областей данных для вывода сообщений имеет следующий вид:

LRIRHRARTCDM

При работе с сообщениями из одной области приоритет имеют меньшие адреса.

При работе с косвенно заданными сообщениями (т.е. \*DM) высший приоритет имеют с меньшими адресами.

**Очистка сообщений**

Для очистки сообщений выполните FAL 00 или очистите с программатора с помощью процедуры 4.6.5.

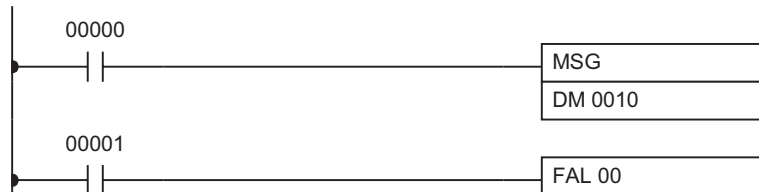
При изменении данных для сообщений изменяются и сообщения.

**Флаги**

**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

**Пример**

Следующий пример показывает индикацию, которая вызывается после включения 00000 в состояние 1 для указанной команды и данных. Если 00001 равно 1, сообщение очищается.



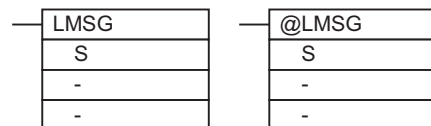
Адрес	Инструкция	Операнд
00000	LD	000 00
00001	MSG	DM 0010
00002	LD	000 01
00003	FAL	00

MSG ABCDEFGHIJKLMN
-----------------------

Содержание DM					Эквивалент ASCII	
DM 0010	4	1	4	2	A	B
DM 0011	4	3	4	4	C	D
DM 0012	4	5	4	6	E	F
DM 0013	4	7	4	8	G	H
DM 0014	4	9	4	A	I	J
DM 0015	4	B	4	C	K	L
DM 0016	4	D	4	E	M	N
DM 0017	4	F	5	0	O	P

**5.26.5 LMSG - Длинное сообщение**

**Обозначение на ПК**



	Номер инструкции	Операнды		
Контроллер	Nr		-	-
			не используется	не используется
C200H-CPU-E	47	IR, SR, HR, AR, LR, TC, DM	000	000
C200H-CPU-ZE	047	IR, SR, HR, AR, LR, TC, DM, EM	000	000

**Ограничения**

S.. S+15 должны лежать в одной области данных и должны быть в ASCII. Сообщение будет обрезано, если между S и S+15 содержится символ нуля (0D).

**Описание**

LMSG служит для вывода сообщения длиной 32 знака на программатор. Сообщение должно быть в коде ASCII, начинаться в S и оканчиваться в S+15 (если только не требуется короче). Более короткое сообщение можно получить, поместив символ нуля (0D) в строку; Символы, следующие далее, выводиться не будут.

Для вывода на программатор он должен быть установлен в режим TERMINAL. Хотя LMSG будет выполняться, как нормальное, сообщение не появится на дисплее, если не задать режим TERMINAL. Подробности о переключения в режим TERMINAL см. 5.25.6.

Когда секция 6 переключателя DIP ЦПУ = OFF, программатор можно переключить в режим TERMINAL клавишей CHG или командой TERM из программы. Когда секция 6 переключателя DIP ЦПУ = ON, программатор можно переключить в режим TERMINAL, включив в 1 бит AR 0709.

**Флаги**

**ER:** S и S+15 не в одной области данных.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

**Пример**

Хотя индикация длиннее и выбирается выходное устройство, кодирование LMSG такое же, как и для MSG. Пример об использовании MSG см. предыдущий пункт.

**5.26.6 TERM - Режим TERMINAL**

**Обозначение на ПКС**

TERM	@TERM
000	000
000	000
000	000

Номер инструкции		Операнды		
Контроллер	№			
C200H□-CPU□□-E	48	000	000	000
C200H□-CPU□□-ZE	048	000	000	000

**Описание**

Когда условия исполнения = 0, TERM не выполняется. Когда условия исполнения = 1, TERM переключает программатор в режим TERMINAL. (Команды MSG, LMSG и функция распределения клавиатуры выполняются в режиме TERMINAL).

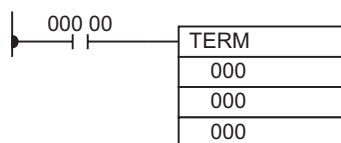
Программатор возвращается в режим CONSOLE при повторном нажатии клавиши CHG. Не существует команд, которые возвращают программатор в режим CONSOLE из программы.

Программатор можно переключить в режим TERMINAL нажатием клавиши CHG на программаторе перед вводом пароля или когда индицируется режим, если только секция 6 переключателя DIP на ЦПУ = OFF. Программатор возвращается в режим CONSOLE при повторном нажатии клавиши CHG

Когда секция 6 переключателя DIP на ЦПУ = ON, программатор можно переключить в режим расширенного TERMINAL, установив в 1 бит AR 0709.

**Пример**

В следующем примере TERM служит для переключения терминала в режим TERMINAL, когда 00000 равен 1. Следите, чтобы секция 6 переключателя DIP на ЦПУ = OFF.



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	TERM	000
		000
		000

### 5.26.7 WDT - Обновление контрольного таймера

#### Обозначение на ПКС

— WDT T — @WDT T

Номер инструкции		Операнды		
Контроллер	Nr	T		
		значение таймера		
C200H□-CPU□□-E	94	# (00..63)		
C200H□-CPU□□-ZE	094	# (00..63)		

#### Описание

Когда условие исполнения = 0, WDT не выполняется. Когда условие исполнения = 1, WDT заносит задание контрольного времени цикла из параметра DM 6618. Значение по умолчанию 120 мс.

Контрольное время = 100 мс x T.

#### Предосторожности

Время контроля цикла (задание контрольного таймера) можно задать от 10 до 640 мс установочными параметрами. Команду WDT можно также использовать для увеличения времени контроля цикла до 640 мс, но увеличивается только часть цикла, используемая для выполнения команд.

WDT можно исполнять более одного раза за цикл, но время цикла нельзя увеличить в сумме более чем до 640 мс. Любое увеличение свыше 640 мс будет игнорироваться. Другая команда WDT не будет выполняться, если время цикла уже увеличено до 640 мс.

Когда время цикла превышает 100 мс, таймеры не будут работать правильно. При использовании WDT один и тот же таймер следует повторять в программе с интервалами менее 100 мс. TIMH нужно использовать только в программе прерывания по расписанию с интервалами 10 мс и менее.

#### Флаги

Данная команда на флаги воздействия не оказывает.

### 5.26.8 IORR - Обновление входов/выходов

#### Обозначение на ПКС

— IORF  
St  
E

Номер инструкции		Операнды		
Контроллер	Nr	St	E	
		первое слово	последнее слово	
C200H□-CPU□□-E	97	IR 000..IR 049, IR 100..IR 199, SR 400..SR 450	IR 000..IR 049, IR 100..IR 199, SR 400..SR 450	
C200H□-CPU□□-ZE	097	IR 000..IR 049, IR 100..IR 199, SR 400..SR 450	IR 000..IR 049, IR 100..IR 199, SR 400..SR 450	

#### Ограничения

IOFR можно использовать для обновления слов входов/выходов, выделенных блокам входов/выходов, специальным блокам и блокам входных прерываний, установленных на ЦПУ или на раме расширения входов/выходов. Команду нельзя использовать для других слов входов/выходов, таких как блоки входов/выходов на ведомых рамах или Модулях группы 2.

**Описание**

Для обновления слов входов/выходов, выделенных ЦПУ или рамкам расширения входов/выходов (IR 000.. IR 029 или IR 300.. IR 309), просто задайте первое (ST) и последнее (E) слово входов/выходов, подлежащих обновлению. Когда условие исполнения для IOFR = 1, все слова от ST до E будут обновлены. Это будет делаться в дополнение к нормальному обновлению входов/выходов, выполняемому во время цикла ПК.

Для обновления слов входов/выходов, выделенных блока специальных входов/выходов 0.. 9 (IR 100.. IR 199), задайте IR 040.. IR 049. Эти слова IR используются для идентификации соответствующего блока специальных входов/выходов. Исполнение IOFR не окажет влияния на содержимое IR 040.. IR 049.

Например, задайте в ST IR 043 и E в IR 045 для обновления слов входов/выходов, выделенным блокам специальных входов/выходов 3,4 и 5. Слова входов/выходов, выделенные данным блокам (IR 130.. IR 159) обновятся при исполнении IOFR. Это будет сделано в дополнение к нормальному обновлению входов/выходов, выполняемому во время цикла ПК.

Для задания конкретной области блоков специальных входов/выходов из 10 слов (IR 100.. IR 190 или IR 400.. IR 450) введите первое слово области из 10 слов. (Последняя цифра слова IR должна быть = 0).

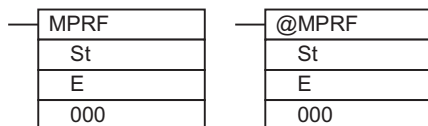
Подробности об обновлении слов, выделенных блокам входов/выходов высокой плотности группы 2, см. 5.25.9.

**Флаги**

- ER:** ST и E вышли за разрешенный диапазон.  
(000.. 029, 040.. 049, 100.. 190, 300.. 309 или 400.. 450).  
ST и E не в одном диапазоне.

**5.26.9 MPRF - Обновление Модулей группы 2**

**Обозначение на ПКС**



Контроллер	Номер инструкции Nr	Операнды		
		St начальный модуль	E конечный модуль	000 не используется
C200H□-CPU□□-E	61	# (0000..000F)	# (0000..000F)	000
C200H□-CPU□□-ZE	061	# (0000..000F)	# (0000..000F)	000

**Ограничения**

MPRF служит для обновления слов входов/выходов, выделенных только блокам входов/выходов высокой плотности группы 2. Команду нельзя использовать для других слов входов/выходов.

ST и E должны лежать в диапазоне #0000.. #000F. ST должно быть меньше или равно E.

**Описание**

Когда условие исполнения = 0, MPRF не выполняется. Когда условие исполнения = 1, слова, выделенные блокам входов/выходов группы 2 с номерами ST.. E будут обновлены. Это произведется в дополнение к нормальному обновлению входов/выходов, которое осуществляется в цикле ЦПУ.

Слова нельзя задавать адресом, а только номером блока входов/выходов, к которым они привязаны.

**Время исполнения**

Время исполнения MPRF подсчитывается следующим образом:

$T_{MPRF}$  = время исполнения команды +(сумма времен обновления входов/выходов блоков группы 2)

Таблицу, в которой указано время обновления входов/выходов блоков группы 2, см. 6.1.

**Флаги**

**ER:** ST и E не являются двоично-десятичными числами в диапазоне #0000.. #000F.  
ST больше E.

**5.26.10 BCNT - Счетчик битов**

**Обозначение на ПКС**

BCNT	@BCNT
N	N
SB	SB
D	D

Контроллер	Номер инструкции №	Операнды		
		N	SB	D
		число слов (BCD)	первое слово блока данных	слово приемник
C200H□-CPU□□-E	67	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM
C200H□-CPU□□-ZE	067	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM

**Ограничения**

N должен быть двоично-десятичным числом в диапазоне 0000.. 6656.

**Описание**

Когда условие исполнения = 0, BCNT не выполняется. Когда условие исполнения = 1, BCNT считает общее количество битов в состоянии 1 во всех словах между SB и SB+(N-1) и помещает результат в D.

**Флаги**

**ER:** N не двоично-десятичное число, или N =0;  
SB и SB+(N-1) находятся в разных области.  
Результат превышает 9999.  
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).  
1, когда значение счетчика (N) превышает границы области данных.  
**EQ:** 1 при результате = 0.

**5.26.11 FCS - Контрольная сумма кадра**

**Обозначение на ПКС**

FCS	@FCS
C	C
R1	R1
D	D

Контроллер	Номер инструкции №	Операнды		
		C	R1	D
		слово состояния	первое слово в блоке	
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, DM, #	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	180	IR, SR, HR, AR, LR, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

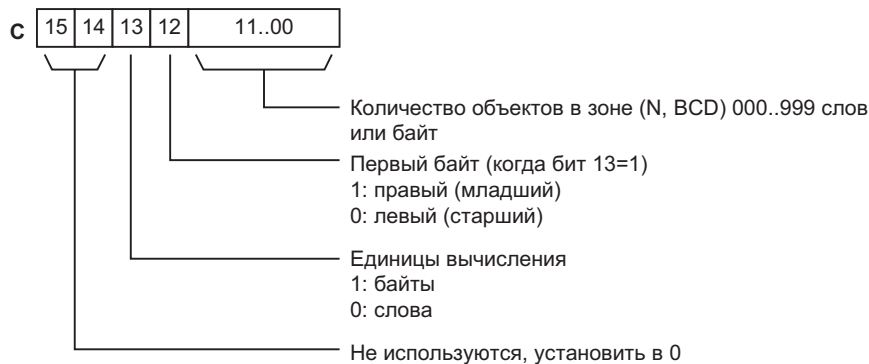
**Описание**

FCS служит для проверки на ошибки при передаче данных через порты связи.  
Когда условие исполнения = 0, FCS не выполняется. Когда условие исполнения = 1, FCS вычисляет контрольную сумму указанной зоны, выполняя ИСКЛЮЧАЮЩЕЕ ИЛИ либо с содержимым слов R1.. R1+N-1 или с содержимым байтов в словах R1.. R1+N-1.



Значение контрольной суммы (16-ричное) преобразуется в ASCII и выдается в слово приемника (D1 и D+1).

Назначение битов в слове управления C показано в следующей диаграмме и объяснено далее.



**Количество объектов в зоне**

Количество объектов в зоне (N) содержится в трех младших цифрах слова C и должно быть двоично-десятичным числом от 001 до 999.

**Единица сложения**

Если бит 13 = 0, подсчитывается контрольная сумма слов;

Если бит 13 = 1, подсчитывается контрольная сумма байт.

Если заданы байты, зона может начинаться либо с младшего, либо старшего байта R1.

Если бит 12 = 0, старший байт R1 не будет включен.

	Старший байт	Младший байт
R1	1	2
R1+1	3	4
R1+2	5	6
R1+3	7	8
..		

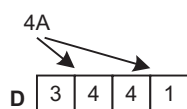
Если бит 12 = 0, над байтами будут производиться операции в следующем порядке: 1, 2, 3, 4..

Если бит 12 = 1, над байтами будут производиться операции в следующем порядке: 2, 3, 4..

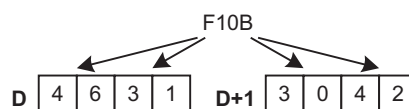
**Преобразование в код ASCII**

Вычисление контрольной суммы байтов дает 2-разрядное 16-ричное значение, которое преобразуется в эквивалент ASCII из 4 цифр. Вычисление контрольной суммы слов дает 4-разрядное 16-ричное значение, которое преобразуется в эквивалент ASCII из 8 цифр, как показано ниже.

Значение контрольной суммы байт



Значение контрольной суммы слов



**Флаги**

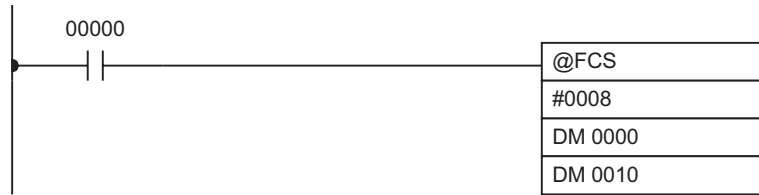
**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

Количество объектов не является двоично-десятичным числом от 001 до 999.

Зона расчета выходит за граница области данных.

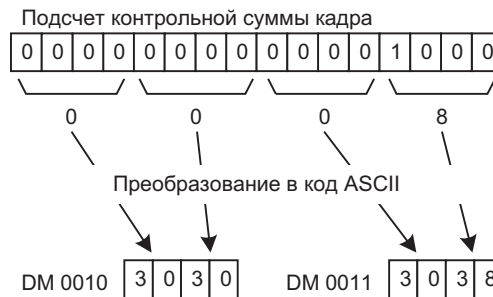
**Пример**

Когда в следующем примере IR 0000 = 1, контрольная сумма блока (0008) вычисляется для 8 слов от DM 0000 до DM 0007 и эквивалент ASCII (30, 30, 30, 38) заносится в DM 0011 и DM 0010.



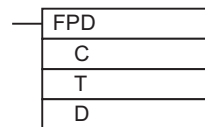
Адрес	Инструкция	Операнд
00000	LD	000 00
00001	@FCS	#0008
		DM 0000
		DM 0010

DM 0000	0001
DM 0001	0002
DM 0002	0003
DM 0003	0004
DM 0004	0005
DM 0005	0006
DM 0006	0007
DM 0007	0008



**5.26.12 FPD - Поиск точки сбоя**

**Обозначение на ПКС**



Контроллер	Номер инструкции №г	Операнды		
		C	T	D
C200H-CPU-E	-	слово состояния	время мониторинга (BCD)	
C200H-CPU-E	-	#	IR, SR, HR, AR, LR, TC, DM, #	IR, HR, AR, LR, DM
C200H-CPU-ZE	269	#	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

**Ограничения**

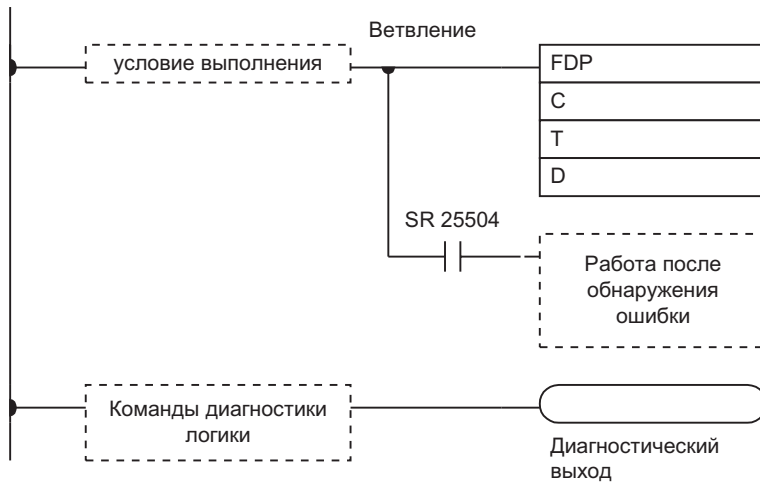
Когда бит 15 слова C = 1, D и D+8 должны лежать в одной области данных

C должна быть введена как константа.

**Описание**

FPD можно использовать в программе столько раз, сколько необходимо, но каждый раз нужно использовать другое значение D. FPD служит для контроля времени между FPD и исполнением диагностического вывода. Если время превышает T, генерируется признак нефатальной ошибки FAL с номером FAL, заданном в C.

Секции программы, отмеченные пунктиром в следующей диаграмме, можно записать по необходимости при особых применениях программы. Обрабатываемая секция программы, переключаемая CY, является дополнительной и может использовать любую команду кроме LD и LD NOT. Команды диагностики логики и условие исполнения может состоять из любой комбинации желаемых условий (нормально открытых и нормально закрытых).

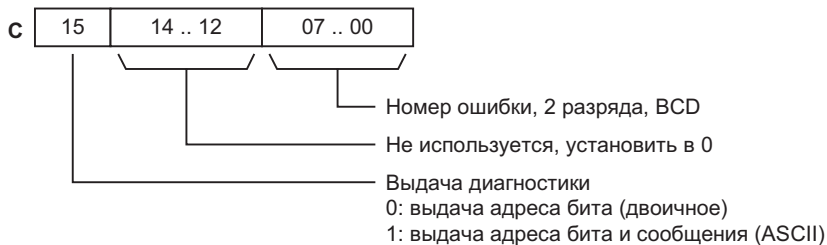


Когда условие исполнения = 0, FPD не выполняется. Когда условие исполнения = 1, FPD контролирует время, пока условие диагностики не установится в 1, и включает в 1 выход диагностики. Если время превышает T, случается следующее:

- 1, 2, 3,... 1. Генерируется признак ошибки FAL с номером FAL, заданном в двух цифрах C. Если задан 00, признак ошибки не генерируется.
2. Команды диагностики логики опрашиваются на первое входное условие = 0, и адрес бита данного условия выдается в слова приемника, начинающихся с D.
3. Флаг CY (SR 25504) = 1. По желанию секция программы обработки признака ошибки может выполняться с использованием флага CY.
4. Если бит 15 слова C = 1, текущее сообщение до 8 знаков ASCII будет индигатироваться на программаторе вместе с адресом бита, упомянутого в шаге 2.

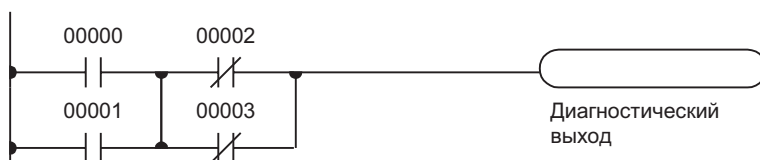
### Слово управления

Функции битов слова управления C показаны на следующей диаграмме.



### Команды диагностики логики

Если время до включения в 1 условия диагностики превысило T, команды диагностики логики ищут входные условия в состоянии 0. Если в состоянии 0 находятся более 1 условия исполнения, выбирается условие на более высокой строке и ближе к левой шине.



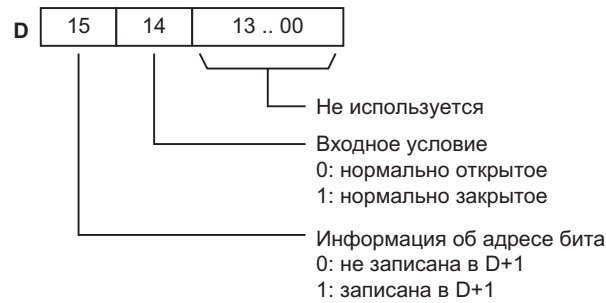
Когда IR 00000.. IR 00003 = 1, в качестве причины невключения в состояние 1 выхода диагностики будет указано нормально открытое условие IR 00002.

### Вывод диагностики

Есть 2 способа вывода адреса условия в состоянии 0, обнаруженного при диагностике.

- 1, 2, 3,... 1. Вывод адреса бита (когда бит 15 слова C = 0).

Бит D.15 указывает, загружена или нет в D+1 информация об адресе бита. Если нет, бит 14 слова D указывает, является ли входное условие нормально открытым или нормально закрытым.

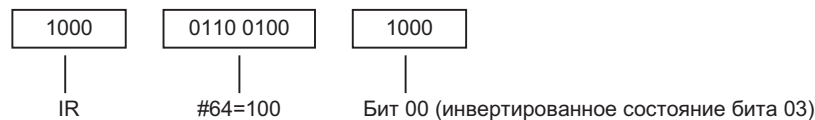


В D+1 содержится код адреса бита входного условия, как показано в таблице. Адреса, номера битов и номера таймеров/счетчиков - в двоичном виде.

Область данных	Состояние бита D+1																	
	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00		
IR, SR (см. прим. 3)	1	0	0	0	Адрес слова								Номер бита					
HR	1	0	0	1	1	Адрес слова								Номер бита				
LR	1	0	0	1	0	0	Адрес слова								Номер бита			
ТС*	1	0	0	1	0	1	*	Номер таймера или счетчика										

- Замечание**
- \* Для области ТС бит 9 слова D+1 указывает, является ли номер номером таймера или счетчика. 0 - таймер, 1 - счетчик.
  - Старший бит в графе номер бита (бит 03) - резерв.
  - Хотя одни и те же адреса слов применяются для обеих зон, бит 13 = 0 для указания IR 00000.. SR 25515 и = 1 для SR 25600.. IR 51115.

Пример: Если D+1 содержит 1000 0110 0100 1000, IR 10000 будет индигироваться следующим образом:



2. Адрес бита и выдача сообщения (выбирается, когда бит 15 слова C = 1).

Бит D.15 указывает, загружена или нет в D+1.. D+3 информация об адресе бита. Если да, бит 14 слова D указывает, является ли входное условие нормально открытым или нормально закрытым. Смотри следующую таблицу.

Слова D+5.. D+8 содержит информацию в ASCII, которая индигируется программатором наряду с адресом бита, когда выполняется FPD. Слова D+5.. D+8 содержит сообщение, предварительно установленное пользователем как показано в следующей таблице.

Слово	Биты 15.. 08	Биты 07.. 00
D+1	20 = пробел	Первый символ ASCII
D+2	Второй символ ASCII	Третий символ ASCII
D+3	Четвертый символ ASCII	Пятый символ ASCII
D+4	2D = "-"	"0"- нормально открыто, "1"- нормально закрыто,
D+5	Первый символ ASCII	Второй символ ASCII
D+6	Третий символ ASCII	Четвертый символ ASCII

D+7	Пятый символ ASCII	Шестой символ ASCII
D+8	Седьмой символ ASCII	Восьмой символ ASCII

**Замечание** Если в сообщении не нужны 8 символов, введите OD после последнего символа.

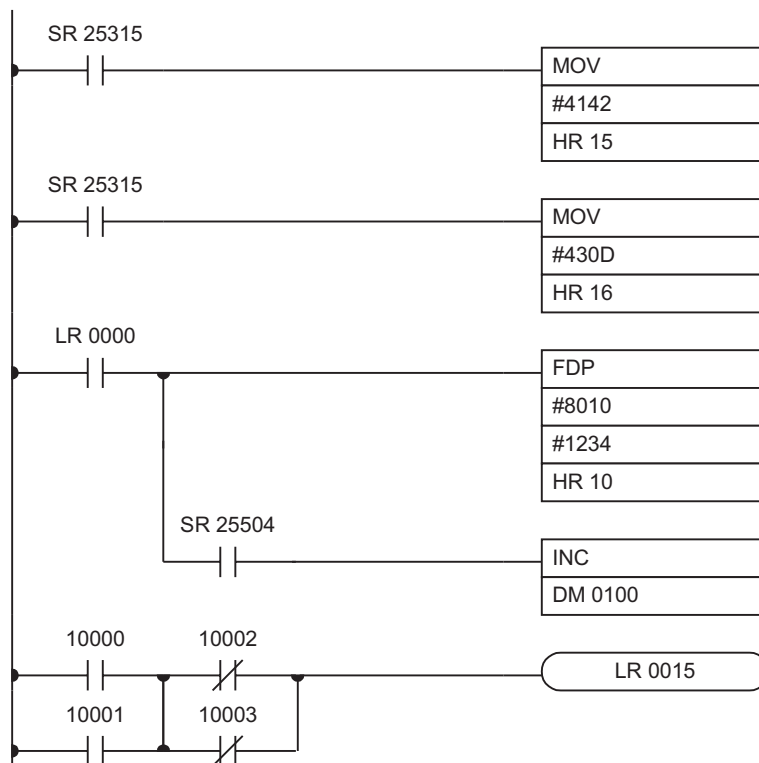
**Задание времени наблюдения**

Далее описанная процедура служит для автоматического задания времени наблюдения, T, в условиях работы при задании словного операнда T.

- 1, 2, 3,... 1. Переключите CQM1 в режим MONITOR.
2. Подключите периферийное устройство, например, программатор.
3. Используйте периферийное устройство для включения бита управление AR 2508.
4. Выполните программу с AR 2508 в положении 1. Если время контроля в T превышено, в T загрузится время контроля, равное 1.5 T. Когда AR 2508 = 1, признак ошибки FAL не выдается.
5. Выключите AR 2508 после того, как в T загрузится приемлемое значение T.

**Пример**

В следующем примере команда FPD задана для индикации адреса бита и сообщение "ABC", когда время контроля превысит 123.4 с.



Адрес	Инструкция	Операнд
00000	LD	253 15
00001	MOV	#4142
		HR 15
00002	LD	253 15
00003	MOV	#430D
		HR 16
00004	LD	LR 0000
00005	FPD	#8010

Адрес	Инструкция	Операнд
		#1234
		HR 10
00006	AND	255 04
00007	INC	DM 0100
00008	LD	100 00
00009	OR	100 01
00010	LD NOT	100 02
00011	OR NOT	100 03
00012	AND LD	
00013	OUT	LR 0015

FPD выполняется и просмотр начинается, когда LR 0000 устанавливается в 1. Если LR 0015 не включается в 1 за 123.4 с и IR 10000.. IR 10003 все в состоянии = 1, IR 10002 будет выбран в качестве причины признака ошибки, будет генерироваться признак ошибки FAL(006) с номером FAL 10, и адрес бита в текущем сообщении ("10002-1ABC") будет индикативаться на периферийном устройстве.

HR 10	0000	HR 10	C000	Указывает на нормально закрытое условие
HR 11	0000	HR 11	2031	'1'
HR 12	0000	HR 12	3030	'00'
HR 13	0000	HR 13	3032	'02'
HR 14	0000	HR 14	2D31	'-1'
HR 15	4142	HR 15	4142	'AB'
HR 16	430D	HR 16	430D	'C' и код возврата каретки
HR 17	0000	HR 17	0000	Последние два пробела игнорируются
HR 18	0000	HR 18	0000	(индицируются как пробелы)

**Флаги**

**ER:** Т не двоично-десятичное число

С не константа или не двоично-десятичное число 00.. 99.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

**СУ:** 1, когда время между исполнением FPD и исполнением выдачи диагностики превышает Т.

**5.26.13 SRCH - Поиск данных**

**Обозначение на ПКС**

SRCH	@SRCH
N	N
R1	R1
C	C

Контроллер	Номер инструкции	Операнды		
		N	R1	C
		число слов	первое слово в блоке	маска, слово результата (C+1)
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	181	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM

**Ограничения**

N должно быть двоично-десятичным числом от 0001 до 9999.

R1 и R1+N-1 должны быть в одной области.

**Описание**

Когда условие исполнения = 0, команда SRCH не выполняется. Когда условие исполнения = 1, команда SRCH ищет в зоне памяти от R1 до R1+N-1 адреса, которые содержат сравниваемые данные в слове C. Если искомые данные содержатся в одном или более словах, флаг EQ (SR 25506) устанавливается в 1 и в C + 1 заносится младший из адресов, содержащих сравниваемые данные. Для области DM адрес задается не так, как для других областей.

- 1, 2, 3,... 1. Для адреса в области DM адрес записывается в C+1. Например, если минимальный адрес, содержащий сравниваемые данные, = DM 0114, тогда #0114 запишется в C+1.
- 2. Для адреса в других областях в C+1 записывается количество адресов от начала поиска. Например, если адрес, содержащий сравниваемые данные, = IR 0114 и первое слово зоны поиска = IR 014, тогда #0100 запишется в C+1.

Если ни один из адресов не содержит сравниваемых данных, флаг EQ (SR 25506) устанавливается в 0 и C1 не меняется.

**Флаги**

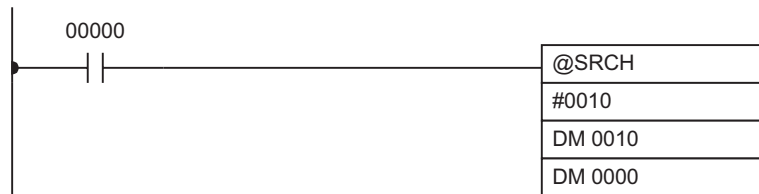
**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

N не двоично-десятичное число в диапазоне 0001.. 6655.

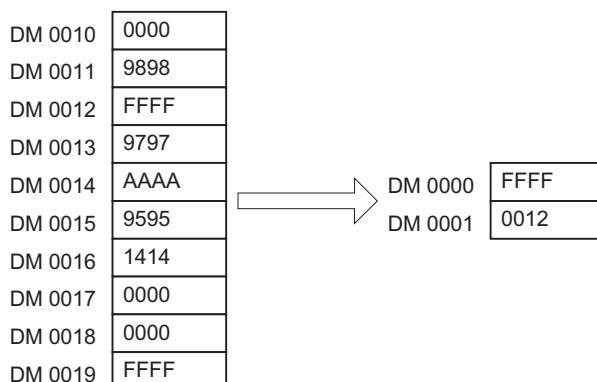
**EQ:** 1, когда сравниваемые данные есть в зоне поиска.

**Пример**

В следующем примере в зоне из 10 слов DM 0010.. DM 0019 ищутся адреса, в которых содержатся одинаковые данные с DM 0000 (#FFFF). Поскольку в DM 0012 содержатся такие же данные, флаг EQ (SR 25506) включается в 1 и в DM 0001 заносится #0012.



Адрес	Инструкция	Операнд
00000	LD	000 01
00001	@SRCH	#0010
		DM 0010
		DM 0000



### 5.26.14 XDMR - Чтение расширенной DM

#### Обозначение на ПКС

XDMR	@XDMR
N	N
S	S
D	D

Контроллер	Номер инструкции №	Операнды		
		N число слов	S первое слово расширенной DM	D
C200H-CPU-E	-	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, DM
C200H-CPU-ZE	280	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, DM, EM

#### Ограничения

N должно быть двоично-десятичным числом от 0001 до 3000.

S должно быть двоично-десятичным числом от 7000 до 9999.

S и S+N-1 должны быть в одной области (как и D и D+N-1).

#### Описание

Когда условие исполнения = 0, команда XDMR не выполняется. Когда условие исполнения = 1, команда XDMR копирует содержимое слов расширенных DM из S.. S+N-1 в слова приемника D.. D+N-1.

#### Предосторожности

Область расширенных DM должна быть задана в установочных параметрах ПК, прежде чем ее можно было использовать в программировании. Не превышайте диапазон области расширенных DM.

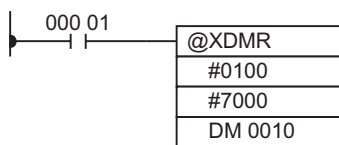
Команде XDMR отдается приоритет при прерывании питания.

#### Флаги

- ER:** Указанная область расширенных DM не существует. Обеспечьте, чтобы указанные слова были выделены расширенным DM. Подробности см. 7.2.15. Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).  
N не двоично-десятичное число в диапазоне 0001.. 3000.  
S не двоично-десятичное число в диапазоне 7000.. 9999.

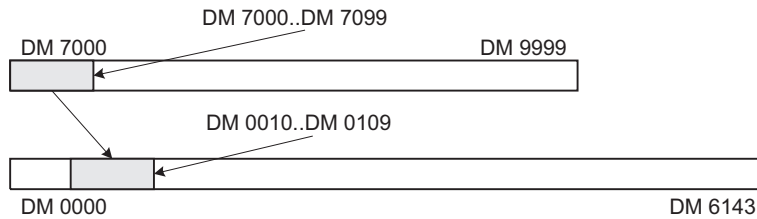
#### Пример

В следующем примере зона из 100 слов DM 7000.. DM 7099 копируется в DM 0010.. DM 0109, когда IR 00001 = 1.



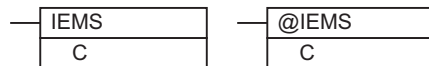
Адрес	Инструкция	Операнд
00000	LD	000 01
00001	@XDMR	#0100
		#7000
		DM 0010





### 5.26.15 IEMS - Косвенная адресация EM

#### Обозначение на ПКС



Номер инструкции		Операнды	
Контроллер	№	C	слово состояния
C200H□-CPU□□-E	-	000, #E000, #E0B1, #E0B2	
C200H□-CPU□□-ZE	-	000, #E000, #E0B1, #E0B2	

#### Ограничения

C должно быть 000, #E000, #E0B0, #E0B1 или #E0B2.

#### Описание

Когда условие исполнения = 1, команда IEMS изменяет адресата косвенной адресации (\*DM) на DM или на указанный банк EM. Текущий номер банка EM можно также изменить, когда косвенная адресация меняется на EM.

Адресат команды \*DM переключается на область DM в начале подпрограммы прерываний. Он также возвращается к области DM в начале каждого сканирования. В следующей таблице показаны позволенные значения для C и их функций.

C	Операция IEMS
000	Переключает адреса та *DM в область DM
#E000	Переключает адресата *DM в текущий банк области EM.
#E0B0	Переключает адресата *DM в банк 0 области EM.
#E0B1	Переключает адресата *DM в банк 1 области EM.
#E0B2	Переключает адресата *DM в банк 2 области EM.

Содержимое DM 6031 указывает на текущего адресата \*DM и текущий банк EM в соответствии с таблицей.

Слово	Биты 00.. 07	Биты 08.. 15
DM 6031	Текущий номер банка EM (00.. 02)	Адресат *DM (00: DM, 01: EM)

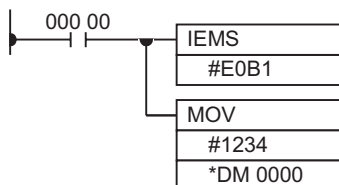
**Замечание** Введите 000 для второго и третьего операнда при использовании команды пересылки.

#### Флаги

**ER:** C имеет недопустимое значение.

#### Пример

В следующем примере IEMS изменяет адресата \*DM на банк 1 области и использует косвенную адресацию для пересылки #1234 в EM 0001 банк 1.



DM 0000	0001	EM 0000	
			1234

### 5.26.16 EMBC - Выбор банка EM

#### Обозначение на РКС

EMBC	@EMBC
N	N

Номер инструкции		Операнды		
Контроллер	№	N		
		номер банка		
C200H□-CPU□□-E		IR, SR, HR, AR, LR, DM, #		
C200H□-CPU□□-ZE		IR, SR, HR, AR, LR, DM, EM, #		

#### Ограничения

N должно быть 0000..000F и банк с заданным номером должен существовать в Модуле ЦПУ.

#### Описание

Когда данная команда выполняется при условии исполнения = 1, EMBS(281) изменяет текущий банк EM на заданный. Модуль ЦПУ может иметь доступ только к текущему банку, который указан в битах 00..07 в DM 6031.

Если указанного банка не существует на Модуле ЦПУ, появится признак ошибки и EMBC(281) не будет исполнена.

#### Флаги

**ER:** Значение слова, в котором находится адрес DM/EM для косвенной адресации не двоично-десятичное число или выходит за границу области DM/EM.

Банк с заданным номером N отсутствует в Модуле ЦПУ.

#### Пример

В следующем примере EMBC(281) изменяет текущий банк на банк номер 2, когда IR 00000 = 1.



### 5.26.17 CMCR - Работа с картой PCMCIA

#### Обозначение на РКС

CMCR	@CMCR
C	C
S	S
D	D

Номер инструкции		Операнды		
Контроллер	№	C	S	D
C200H□-CPU□□-E				
C200H□-CPU□□-ZE	261	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM

#### Ограничения

DM 6144..DM 6655 нельзя использовать для D.

#### Описание

Когда условие исполнения = 0, CMCR(261) не выполняется. Когда условие исполнения = 1, CMCR(261) выполняет макропроцесс, который пишет, читает, сравнивает или производит поиск в карте Модуля интерфейса PCMCIA Card.

Когда CMCR(261) записывает данные в файл с разделителем-запятой, этот файл можно записать целиком одной строкой. Файлы с разделителем-запятой, которые были созданы на персональном компьютере, считаются записанными в одном из следующих форматов. (Не имеет значения, является ли запятая отдельным байтом).

4байта, запятая, 4 байта, запятая, 4байта, запятая..

8 байт, запятая, 8 байт, запятая, 8байт, запятая..

Файл может быть дополнен только если есть свободное место на Карте и данные можно переписать только когда задано приемлемое смещение.

Поля в файле с разделителем-запятой не должны быть ограничены разделителями, такими как знаки препинания.

**Блок параметров управления**

Записывайте данные в блок параметров управления используя формат, показанный на следующем рисунке.



Введите имя дисковода и файла (в ASCII), который предполагается записать, считать, сравнить или найти. Обязательно включайте все три буквы расширения файла. Можно задавать директории, если они созданы. Слот 1 - дисковод 'G', слот 2 - дисковод 'H'

**Номер процесса**

Номер процесса (1..4) задает, какая из функций CMCR будет выполняться.

Номер процесса	Имя процесса	Функция
1	Запись файла	Записывает данные из памяти ПК в указанный файл Карты Модуля PC Card
2	Чтение файла	Читает данные из указанного файла Карты Модуля PC Card в память ПК.
3	Сравнение файла с памятью	Сравнивает указанный файл Карты Модуля PC Card с данными в памяти ПК.
4	Поиск файла	Ищет указанный файл на Карте Модуля PC Card

**Настройки порта**

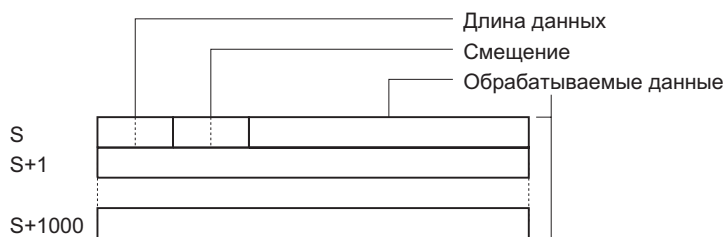
Установочные параметры порта (биты 8..10) задают конкретные детали операций CMCR(261) и формата файлов PC Card.

Номер процесса	Настройки порта		
	Бит 8	Бит 9	Бит 10
1	0: Переписать существующий файл	0: Создать файл с разделителем-запятой	0: Разделитель через одно слово
	1: Создать новый файл	1: Создать двоичный файл	1: Разделитель через два слова

Номер процесса	Настройки порта		
	Бит 8	Бит 9	Бит 10
2	0: Читать заданный файл 1: Читать число элементов в заданном файле	0: Читать как файл с разделителем-запятой 1: Читать как двоичный файл	0: Разделитель через одно слово 1: Разделитель через два слова
3	-	0: Сравнить как файл с разделителем-запятой 1: Сравнить как двоичный файл	0: Разделитель через одно слово 1: Разделитель через два слова
4	-	0: Искать как файл с разделителем-запятой 1: Искать как двоичный файл	0: Разделитель через одно слово 1: Разделитель через два слова

**Блок обрабатываемых данных**

Блок обрабатываемых данных включает параметры «длина данных», смещение и сами обрабатываемые данные. Для параметра «длина данных» задавайте длину обрабатываемых данных +1. Максимальная длина обрабатываемых данных = 1000 слов.



Настройки параметров «длина данных», смещение и самих данных зависят от номера процесса в соответствии со следующей таблицей.

Номер процесса	Длина данных	Смещение	Обрабатываемые данные
1	Число слов (двоично-десятичное число: 1..1001)	Число элементов записываемых данных (0..FFFF) Задавайте число слов для файла с разделителем через одно слово и двоичного файла. Задавайте число элементов для файла с разделителем через два слова.	Данные, которые должны быть записаны в файл. Макс. 999 слов для двоичного файла и с разделителем через одно слова. Макс. 999 слов (449 элементов) для файла с разделителем через два слова.
2	Всегда задается 0003	Число элементов читаемых данных (0..FFFF) Задавайте число слов для файла с разделителем через одно слово и двоичного файла. Задавайте число элементов для файла с разделителем через два слова.	Задавайте число слов для чтения (16-ричное число 1..3E7) Макс. 999 (\$3E7) слов для двоичного файла и с разделителем через одно слово. Макс. 449 (\$1F3) слов для файла с разделителем через два слова.

Номер процесса	Длина данных	Смещение	Обрабатываемые данные
3	Число слов (двоично-десятичное число: 1..1001)	Число элементов для сравнения (0..FFFF) Задавайте число слов для файла с разделителем через одно слово и двоичного файла. Задавайте число элементов для файла с разделителем через два слова.	Данные для сравнения. Макс. 999 слов для двоичного файла и с разделителем через одно слова. Макс. 999 слов (449 элементов) для файла с разделителем через два слова.
4	Задавайте 3 для файла с разделителем через одно слова и двоичного файла. Задавайте 4 для файла с разделителем через два слова.	Число элементов для поиска (0..FFFF) Задавайте число слов для файла с разделителем через одно слово и двоичного файла. Задавайте число элементов для файла с разделителем через два слова.	Данные для сравнения. Задавайте одно слово данных для поиска для двоичного файла и с разделителем через одно слово. Задавайте два слова данных для поиска для файла с разделителем через два слова.

**Замечание** 1, Когда смещение = 0 задано для процесса № 1 и уже существует файл с таким же именем, существующий файл будет удален и создан новый файл. Если длина данных установлена = 1 (нет данных для записи), команда просто удалит существующий файл. Если смещение задано = 1, данные будут добавлены в конец существующего файла.

2. Файл с разделителем-запятой - это файл, в котором однобайтный символ (такой, как запятая) вставлен после каждых 4 или 8 байт данных. Ошибка (код завершения 2) появится, если имеются 2-байтные разделители, такие как CR+LF. Предварительно замените эти 2-байтные разделители на 1-байтные разделители.

#### Результаты ответа

Результаты ответа зависят от заданного номера процесса в соответствии со следующей таблицей.

Номер процесса	Результаты ответа
1	Нет
2	Содержит данные, считанные из заданного файла. Когда считано число элементов, оно содержится в виде 2 - разрядного 16-ричного слова.
3	Нет (Результат сравнения возвращен в коде окончания)
4	Когда искомые данные найдены в файле, их местоположение возвращается как число слов или элементов данных (0..FFF) от местоположения смещения. Начало файла: 0000 Второе слово (элемент): 0001 Третье слово (элемент): 0002

#### Биты SR и код окончания

Код окончания команды выдается в SR 237 после выполнения CMCR(261). Кроме того, в SR 252 хранятся флаги, показывающие состояние завершения команды (нормальное/с ошибкой) и состояние исполнения для уровней 0 и 1. В следующей таблице показаны функции этих битов.

Слово	Бит(ы)	Функция
SR 237	00..07	Область выдачи кода окончания для уровня 0 после исполнения CMCR.
	08..15	Область выдачи кода окончания для уровня 1 после исполнения CMCR.
SR 252	00	Флаг ошибки уровня 0 после исполнения CMCR.
	01	1, когда CMCR(261) можно исполнять для уровня 0

Слово	Бит(ы)	Функция
	02	Флаг ошибки уровня 1 после исполнения CMCR(261).
	03	1, когда CMCR(261) можно исполнять для уровня 1

В следующей таблице показаны значения кодов окончания.

Код	Значение
00	Нормальное завершение
01	Ошибка параметра, такого как смещение, размер файла или число слов для чтения.
02	Диск полон, ошибка ввода/вывода файла или ошибка типа файла
03	Задан несуществующий файл
04	Сбой поиска или сравнения
05..FE	Не определено
FF	Ошибка номера процесса

### Флаги

**ER:** Значение слова, в котором находится адрес DM/EM для косвенной адресации не двоично-десятичное число или выходит за границу области DM/EM.

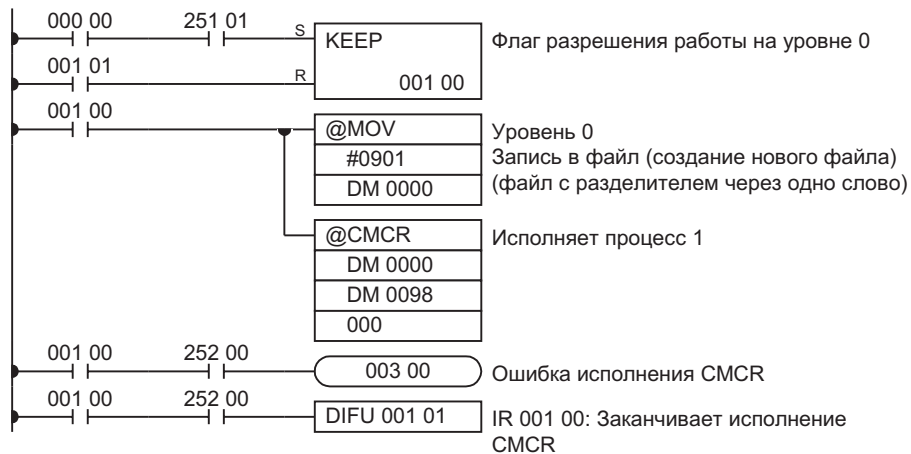
Для D использовались DM 6144..DM 6655.

Флаг разрешения исполнения на заданном уровне (SR 25201 или SR 25204) = 0.

Задаваемый уровень был не 0 или 1.

### Пример

При переключения IR 00000 0 - 1 - 0 100 слов области DM, начиная от DM 0100, переписываются в файл G:\DMSAVE.DAT (Карта памяти, платоместо 1). Настройки области параметров показаны в таблице после релейно-контактной схемы.



В DM 0000..DM 0007 содержатся параметры управления, а в DM 0098..DM 0199 содержатся обрабатываемые данные в соответствии со следующей таблицей..

Слово	Значение	Функция
DM 0000	---	Управляющие параметры
DM 0001	47 3A	ASCII: «G:»
DM 0002	5C 44	ASCII: «\D»
DM 0003	4D 53	ASCII: «MS»
DM 0004	41 56	ASCII: «AV»
DM 0005	45 2E	ASCII: «E.»
DM 0006	44 41	ASCII: «DA»
DM 0007	54	ASCII: «T»

## 5.26 Специальные команды

---

Слово	Значение	Функция
DM 0098	0102	Длина записываемых данных: 102 (16-ричное число)
DM 0099	0000	Смещение: 0
DM 0100	-	Данные для записи
..		
DM 0199	-	Данные для записи

## 5.27 Команды связи

Команды связи используются для связи с другими ПК, блоками BASIC или управляющими компьютерами, связанными по системе SYSMAC NET, SYSMAC LINK или ETHERNET.

### 5.27.1 SEND - Передача по сети

#### Обозначение на ПКС

SEND	@SEND
S	S
D	D
C	C

Контроллер	Номер инструкции Nr	Операнды		
		S	D	C
C200H□-CPU□□-E	90	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM
C200H□-CPU□□-ZE	090	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM

#### Ограничения

S..C+2 должны находиться в одной области данных и должны лежать в диапазонах, описанных далее. Для использования SEND в системе должна быть связь SYSMAC LINK, SYSMAC NET или блок PC CARD.

#### Описание

Когда условие исполнения = 0, команда SEND не выполняется. Когда условие исполнения = 1, команда SEND передает данные, начинающиеся в слове A по адресу, указанному в B, на заданный узел системы SYSMAC LINK, SYSMAC NET или ETHERNET. Для слов управления, начинающихся с C, задайте количество посылаемых слов, узел адресата и другие параметры. Содержимое параметров управления зависит от системы, по которой осуществляется передача: SYSMAC LINK, SYSMAC NET или ETHERNET.

Состояние бита 15 слова C+1 определяет, для каких систем используется команда: SYSMAC NET или SYSMAC LINK/ETHERNET.

#### Параметры управления

##### Система ETHERNET

Для отправки данных по всем узлам задайте номер узла адресата 0. Подробности см. руководство по блоку PC CARD.

Слово	Биты 00..07	Биты 08..15
C	Количество слов (0..1000, 4-разрядные шестнадцатиричные, например, 0000 <sub>16-рич.</sub> 03E8 <sub>16-рич.</sub> )	
C+1	Предельное значение времени ответа (0.1..25.5 секунд с дискретой задания 0.1 с, 2-разрядные 16-ричные без десятичной точки, например, 01 <sub>16-рич.</sub> FF <sub>16-рич.</sub> )	Биты 08..11: Число повторов (0..15 16-ричных, например, 0 <sub>16-рич.</sub> F <sub>16-рич.</sub> Бит 12: 1: Косвенная адресация, 0: Прямая адресация Бит 13: 1: Ответ не возвращается, 0: Ответ возвращается Бит 14: 1: Уровень 0, 0: Уровень 1 Бит 15: Установлен в 1
C+2	Узел адресата (0..127, 2-разрядные 16-ричные, например, 00 <sub>16-рич.</sub> 7E <sub>16-рич.</sub> )*	Биты 08..12: Адрес блока узла адресата. Задайте в 00 <sub>16-рич.</sub> Биты 13..15: Задайте = 0.

##### Система SYSMAC NET

Номер порта адресата всегда задан в 0. Задавайте номер узла адресата в 0 для отправки данных по всем узлам. Задайте номер сети в 0 для отправки данных в узел в той же подсистеме (т. е. сети). Подробности см. руководство по SYSMAC NET.



Слово	Биты 00..07	Биты 08..15
C	Количество слов (0..1000, 4-разрядные шестнадцатиричные, например, 0000 <sub>16-рич.</sub> 03E8 <sub>16-рич.</sub> )	
C+1	Номер сети (0..127, 2-разрядные 16-ричные, например, 00 <sub>16-рич.</sub> 7F <sub>16-рич.</sub> )*	Бит 14: 1: Уровень 0, 0: Уровень 1 Биты 08..13 и 15: Установить в 0.
C+2	Узел адресата (0..126, 2-разрядные 16-ричные, например, 00 <sub>16-рич.</sub> 7E <sub>16-рич.</sub> )*	Порт адресата NSB: 00 NSU: 01/02

**Замечание** \*Можно задать номер узла ПК, производящего посылку.

**Система SYSMAC LINK**

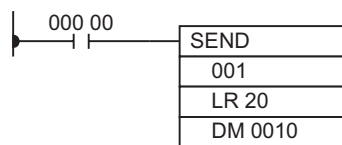
Для посылки данных по всем узлам задайте номер узла адресата 0. Подробности см. руководство по SYSMAC LINK.

Слово	Биты 00..07	Биты 08..15
C	Количество слов (0..1000, 4-разрядные шестнадцатиричные, например, 0000 <sub>16-рич.</sub> 03E8 <sub>16-рич.</sub> )	
C+1	Предельное значение времени ответа (0.1..25.4 секунд, 2-разрядные 16-ричные без десятичной точки, например, 01 <sub>16-рич.</sub> FF <sub>16-рич.</sub> ) Прим.: Предельное значение времени ответа будет равно 2с, если предел задан на 0 <sub>16-рич.</sub> . Если предел задан FF <sub>16-рич.</sub> , ограничения времени не будет.	Биты 08..11: Число повторов (0..15 16-ричных, например, 0 <sub>16-рич.</sub> F <sub>16-рич.</sub> ) Бит 12: Установить в 0 Бит 13: 1: Ответ не возвращается, 0: Ответ возвращается Бит 14: 1: Уровень 0, 0: Уровень 1 Бит 15: Установить в 1
C+2	Узел адресата (0..62, 2-разрядные 16-ричные, например, 00 <sub>16-рич.</sub> 3E <sub>16-рич.</sub> )*	Установить в 0

**Замечание** \*Нельзя задать номер узла ПК, производящего посылку.

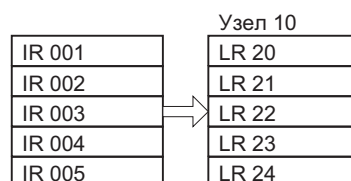
**Примеры**

Данный пример для системы SYSMAC NET. Когда 00000 = 1, следующая программа передает содержимое IR 000.. IR 005 в LR 20..LR 24 на узле 10.



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	SEND	001
		LR 20
		DM 0010

	15	0
DM 0010	0	5
DM 0011	0	0
DM 0012	0	A



**Флаги**

**ER:** Заданный номер узла превышает 126 в SYSMAC NET, 62 в SYSMAC LINK или 127 в ETHERNET.

Посылаемые данные переходят границу допустимой зоны.

Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

Отсутствуют SYSMAC NET/ SYSMAC LINK или блок PC CARD.

## 5.27.2 RECV - Прием из сети

## Обозначение на ПКС

RECV	@RECV
S	S
D	D
C	C

	Номер инструкции	Операнды		
Контроллер	№			
C200H□-CPU□□-E	98	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM
C200H□-CPU□□-ZE	098	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM

## Ограничения

C..C+2 должны находиться в одной области данных и должны лежать в диапазонах, описанных далее. Для использования RECV в системе должна быть смонтирована связь SYSMAC LINK, SYSMAC NET или блок PC CARD.

## Описание

Когда условие исполнения = 0, команда RECV не выполняется. Когда условие исполнения = 1, команда RECV передает данные, начинающиеся в слове S из узла системы SYSMAC LINK, SYSMAC NET или ETHERNET в слова, начинающиеся с D. Для слов управления, начинающихся с C, задайте количество принимаемых слов, узел источника и другие параметры.

Состояние бита 15 слова C+1 определяет, для каких систем используется команда: SYSMAC NET или SYSMAC LINK/ETHERNET.

## Параметры управления

## Система ETHERNET

Подробности см. руководство по блоку PC CARD.

Слово	Биты 00..07	Биты 08..15
C	Количество слов (0..1000, 4-разрядные шестнадцатиричные, например, 0000 <sub>16-рич.</sub> 03E8 <sub>16-рич.</sub> )	
C+1	Предельное значение времени ответа (0.1..25.5 секунд с дискретой задания 0.1 с, 2-разрядные 16-ричные без десятичной точки, например, 01 <sub>16-рич.</sub> FF <sub>16-рич.</sub> ). По умолчанию 00 <sub>16-рич.</sub> (2.2 секунд).	Биты 08..11: Число повторов (0..15 16-ричных, например, 0 <sub>16-рич.</sub> F <sub>16-рич.</sub> Бит 12: 1: Косвенная адресация, 0: Прямая адресация Бит 13: 1: Ответ не возвращается, 0: Ответ возвращается Бит 14: 1: Уровень 0, 0: Уровень 1 Бит 15: Установить в 1
C+2	Узел адресата (0..127, 2-разрядные 16-ричные, например, 00 <sub>16-рич.</sub> 7E <sub>16-рич.</sub> )*	Биты 08..12: Адрес блока узла источника. Задайте в 00 <sub>16-рич.</sub> Биты 13..15: Задайте = 0.

## Система SYSMAC NET

Номер порта источника всегда задан в 0. Задайте номер сети 0 для приема данных узла той же самой подсистемы (т. е. сети). Подробности см. руководство по SYSMAC NET.

Слово	Биты 00..07	Биты 08..15
C	Количество слов (0..1000, 4-разрядные шестнадцатиричные, например, 0000 <sub>16-рич.</sub> 03E8 <sub>16-рич.</sub> )	
C+1	Номер сети (0..127, 2-разрядные 16-ричные, например, 00 <sub>16-рич.</sub> 7F <sub>16-рич.</sub> )*	Бит 14: 1: Уровень 0, 0: Уровень 1 Биты 08..13 и 15: Установить в 0.
C+2	Узел адресата (0..126, 2-разрядные 16-ричные, например, 00 <sub>16-рич.</sub> 7E <sub>16-рич.</sub> )*	Порт источника NSB: 00 NSU: 01/02

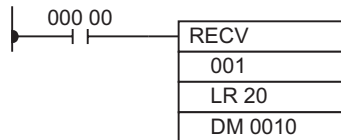
**Система SYSMAC LINK**

Подробности см. руководство по SYSMAC LINK.

Слово	Биты 00..07	Биты 08..15
C	Количество слов (0..1000, 4-разрядные шестнадцатиричные, например, 0000 <sub>16-рич.</sub> 03E8 <sub>16-рич.</sub> )	
C+1	Предельное значение времени ответа (0.1..25.4 секунд, 2-разрядные 16-ричные без десятичной точки, например, 00 <sub>16-рич.</sub> ..FF <sub>16-рич.</sub> ) Прим.: Предельное значение времени ответа будет равно 2с, если предел задан на 0 <sub>16-рич.</sub> . Если предел задается FF <sub>16-рич.</sub> , ограничения времени не будет.	Биты 08..11: Число повторов (0..15 16-ричных, например, 0 <sub>16-рич.</sub> ..F <sub>16-рич.</sub> ) Бит 12: Установить в 0. Бит 13: Установить в 0. Бит 14: 1: Уровень 0, 0: Уровень 1 Бит 15: Установить в 1
C+2	Узел адресата (0..62, 2-разрядные 16-ричные, например, 00 <sub>16-рич.</sub> ..3E <sub>16-рич.</sub> )*	Установить в 0

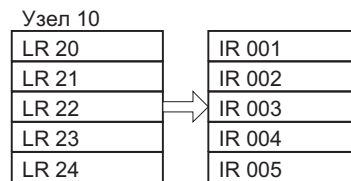
**Примеры**

Данный пример для системы SYSMAC LINK. Когда 00000 = 1, следующая программа передает содержимое IR 000.. IR 005 в LR 20..LR 24 на узле 10.



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	RECV	001
		LR 20
		DM 0010

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DM 0010	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5
DM 0011	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DM 0012	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A



**Флаги**

- ER:** Заданный номер узла превышает 126 в SYSMAC NET, 62 в SYSMAC LINK или 127 в ETHERNET.  
Принимаемые данные переходят границу допустимой зоны.  
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).  
Отсутствуют SYSMAC NET/ SYSMAC LINK или блок PC CARD.

**5.27.3 Общие сведения о сетях**

SEND и RECV основаны на процессах запрос/ответ. То есть, передача не окончена, пока посылающий узел не получит подтверждения с принимающего узла. Обратите внимание, что флаг разрешения SEND/RECV не включается в 1 до первого END после завершения передачи. Подробности об операциях запрос/ответ см. инструкцию по SYSMAC NET или SYSMAC LINK.

Если применяется несколько операций SEND/RECV, перед началом следующей операции передачи/приема SEND/RECV, нужно использовать флаги, указанные в таблице, для того, чтобы установить, завершилась ли предыдущая операция.

Флаг SR	Функции
Флаги разрешения SEND/RCV (SR 25201, SR 25204)	0 во время выполнения операций SEND/ RECV (включая обработку ответа на запрос). Не начинайте операцию SEND/ RECV, пока флаг не установится в 1.
Флаги ошибки SEND/RCV (SR 25200, SR 25203)	0 после нормального завершения SEND/ RECV, т.е. после приема сигнала ответа). 1 при неудачной попытке SEND/RCV. Состояние ошибки сохраняется до следующей операции SEND/RCV. Типы ошибок: Ошибка ВРЕМЯ ИСТЕКЛО (время запрос/ответ больше 1 с) Ошибка передачи данных.

**Синхронизация по времени**

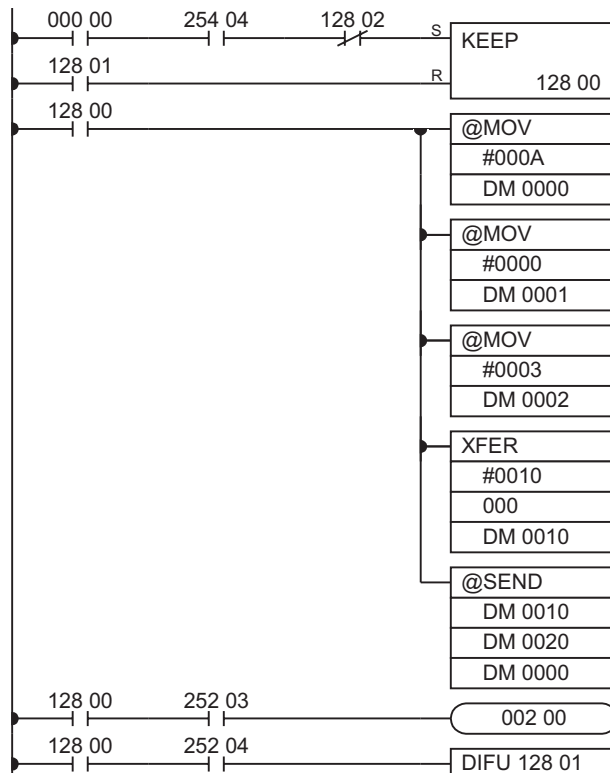


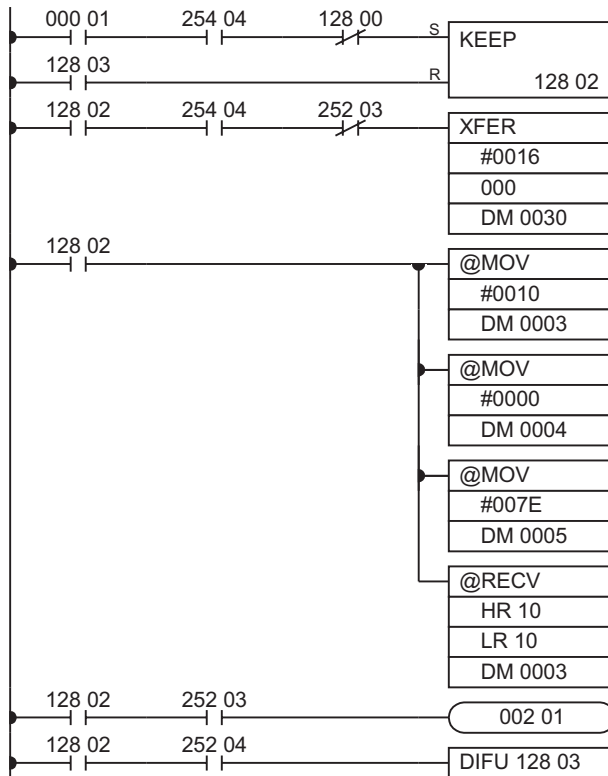
**Обработка данных для SEND/RCV.**

Данные передаются для SEND и RCV для всех ПК, когда исполняются команды SEND/RCV. Финальная обработка передачи/приема осуществляется во время обслуживания периферийных устройств и блоков связи.

**Пример программы: Несколько SEND/RCV.**

Для обеспечения успешного исполнения SEND/RCV Ваша программа должна использовать флаги разрешения SEND/RCV и флаги ошибок SEND/RCV для подтверждения того, что исполнение возможно. Следующая программа показывает пример, как это делать для SYSMAC NET.





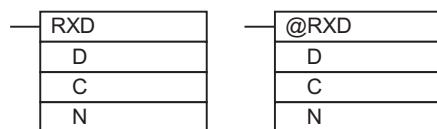
Адрес	Инструкция	Операнд
00000	LD	000 00
00001	AND	252 04
00002	AND NOT	128 02
00003	LD	128 01
00004	KEEP	128 00
00005	LD	128 00
00006	@MOV	#000A
		DM 0000
00007	@MOV	#0000
		DM 0001
00008	@MOV	#0003
		DM 0002
00009	@XFER	#0010
		000
		DM 0010
00010	@SEND	DM 0010
		DM 0020
		DM 0000
00011	LD	128 00
00012	AND	252 03
00013	OUT	002 00
00014	LD	128 00
00015	AND	252 04
00016	DIFU	128 01
00017	LD	000 01
00018	AND	252 04

Адрес	Инструкция	Операнд
00019	AND NOT	128 00
00020	LD	128 03
00021	KEEP	128 02
00022	LD	128 02
00023	AND	252 04
00024	AND NOT	252 03
00025	XFER	#0016
		000
		DM 0030
00026	LD	128 02
00027	@MOV	#0010
		DM 0003
00028	@MOV	#0000
		DM 0004
00029	@MOV	#007E
		DM 0005
00030	@RECV	HR 10
		LR 10
		DM 0003
00031	LD	128 02
00032	AND	252 03
00033	OUT	002 01
00034	LD	128 02
00035	AND	252 04
00036	DIFU	128 03

## 5.28 Команды последовательной связи

## 5.28.1 RXD - Прием

## Обозначение на ПКС



Контроллер	Номер инструкции №г	Операнды		
		D	C	N
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #
C200H□-CPU□□-ZE	235	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #

## Ограничения

D и D+(N\_2)-1 должны быть в одной области

N должно быть двоично-десятичным числом от #0000.. #0256.

## Описание

Когда условие исполнения = 0, команда RXD не выполняется. Когда условие исполнения = 1, команда RXD считывает N байтов данных, принятых с порта, заданного в параметре управления, и затем записывает данные в слова D.. D+(N\_2)-1. За один раз можно читать до 256 байт данных.

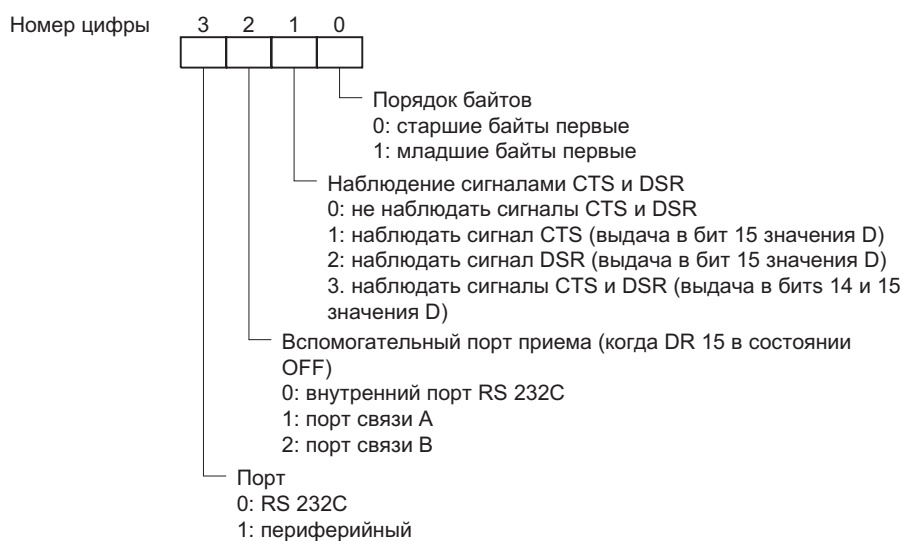
Если принято меньше, чем N байт, будут считаны принятое количество байтов.

**Замечание** RXD требуется для принятия данных только от периферийного порта или порта RS-232C. Передача от управляющего компьютера на блок HOST LINK обрабатывается автоматически и не требует программирования.

**Внимание!** ПК не сможет больше принимать данные, если полученные 256 байтов данных не прочитаны командой RXD. Считывайте данные как можно быстрее после установки в 1 Флага Завершения Приема (SR 26414 для периферийного порта и SR 26406 для порта RS-232C).

## Слово управления

Значение слова управления определяет порт, с которого читаются данные и порядок, в котором данные будут записываться в память.



Порядок, в котором данные записываются в память, зависит от значений цифры 0 в слове C. Восемь байтов данных 12345678.. будут записаны следующим образом:

Цифра 0 = 0			Цифра 0 = 0		
	MSB	LSB		MSB	LSB
D	1	2	D	2	1
D+1	3	4	D+1	4	3
D+2	5	6	D+2	6	5
D+3	7	8	D+3	8	7
..					

**Замечание** MSB - старшие байты, LSB - младшие байты

**Флаги**

- ER:** В ЦПУ нет порта RS-232C.  
 К указанному порту не подключено ответное устройство.  
 Ошибка в задании установочных параметров связи или оперативных параметров.  
 Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).  
 Слова приемника (D.. D+(N<sub>2</sub>)-1) вышли за пределы области данных.

**Периферийный порт**

- 26 414:** SR 26414 установится в 1, когда данные приняты нормально на периферийный порт и сбрасывается после чтения данных командой RXD.  
**266:** SR 266 содержит число байтов, принятых на периферийный порт и сбрасывается в 0 после исполнения RXD.

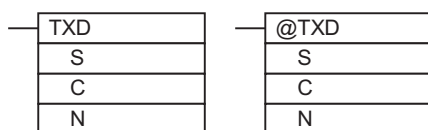
**Порт RS-232C**

- 26 406:** SR 26406 установится в 1, когда данные приняты нормально на порт RS-232C и сбрасывается после чтения данных командой RXD.  
**265:** SR 265 содержит число байтов, принятых на порт RS-232C и сбрасывается в 0 после исполнения RXD.

**Замечание** Флаги и счетчики связи можно очистить либо заданием 0000 для N, либо использованием биты сброса порта (SR 25208 для периферийного порта и SR 25209 для порта RS-232C).

5.28.2 TXD - Передача

**Обозначение на ПКС**



Контроллер	Номер инструкции	Операнды		
		S	C	D
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #
C200H□-CPU□□-ZE	236	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #

**Ограничения**

- S и S+(N<sub>2</sub>)-1 должны быть в одной области  
 N должно быть двоично-десятичным числом от #0000.. #0256. (#0000.. #0061 в режиме связи с верхним уровнем HOST LINK).

**Описание**

Когда условие исполнения = 0, команда TXD не выполняется. Когда условие исполнения = 1, команда TXD считывает N байтов данных из слов S до S+(N<sub>2</sub>)-1, преобразует в ASCII и выдает данные на заданный порт. TXD работает в режиме связи с верхним



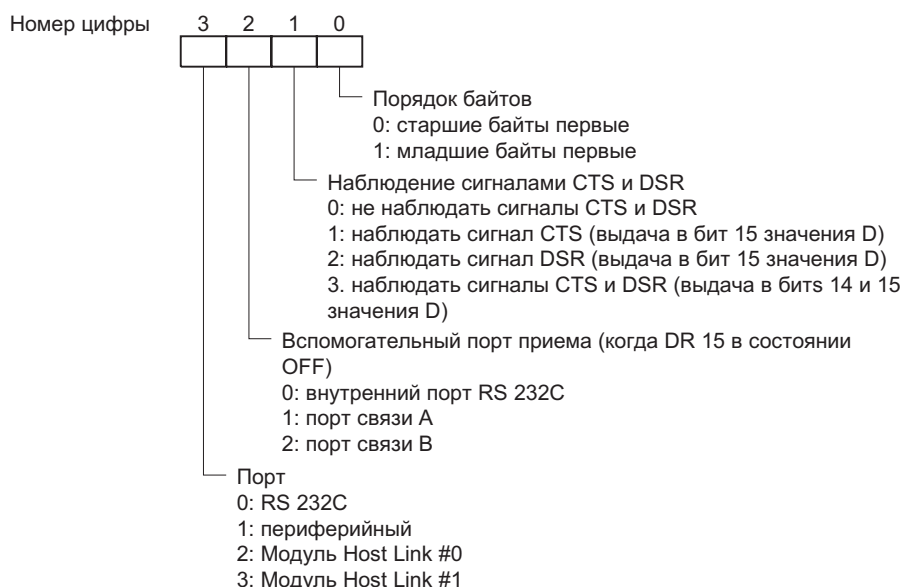
уровнем и режиме RS-232C по-разному, поэтому данные режимы рассмотрены отдельно.

**Замечание** Следующие флаги будут = 1 для указания того, что возможна связь с различными портами. Перед выполнением TXD убедитесь, что соответствующий флаг = 1.

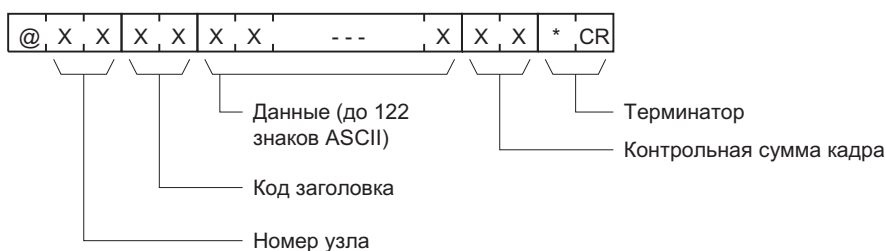
- SR 26405: порт RS-232C
- SR 26413: Периферийный порт
- SR 26705: Блок HOST LINK #0
- SR 26713: Блок HOST LINK #1

**Режим связи с верхним уровнем HOST LINK**

N должно быть двоично-десятичным числом от #0000 до #0061 (т.е. до 122 байт ASCII). Значение параметра управления задает порт, с которого должны передаваться данные, как показано ниже.



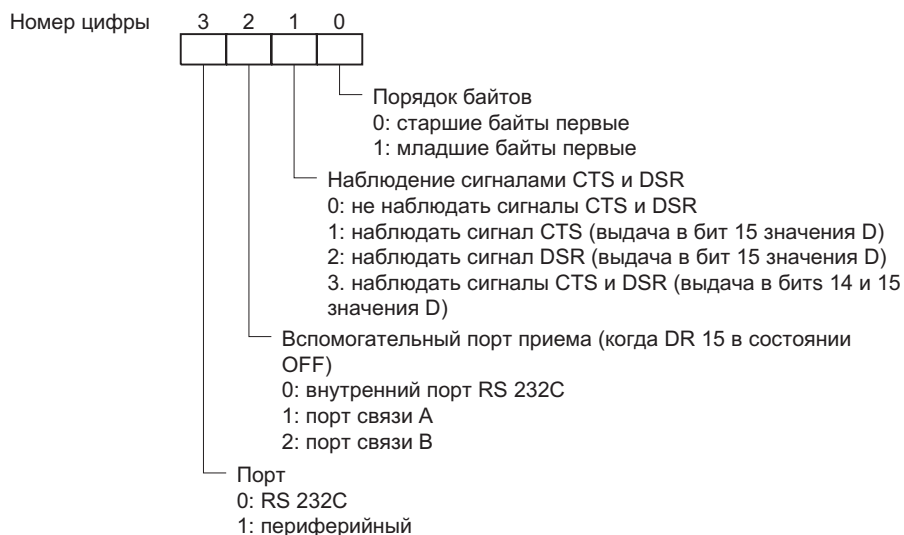
Заданное количество байт будет считано с S.. S+(N\_2)-1, преобразовано в ASCII и выдано через заданный порт. Байты источника, приведенные в таблице, будут переданы в порядке: 12345678..



На следующем рисунке показан формат команды связи с верхним уровнем (TDX), посланный с ПК. В зависимости от установочных параметров C200HX/HG/HE и C200HS автоматически пристыковывают префиксы и суффиксы, такие как номер узла, заголовок и контрольную сумму блока.

**Режим RS-232C**

N должно быть двоично-десятичным числом от #0000 до #0256. Значение параметра управления задает порт, с которого должны передаваться данные, и порядок, в котором данные будут записаны в память.



Заданное количество байт будет считано с S до S+(N\_2)-1, преобразовано в ASCII и выдано на заданный порт.

	MSB	LSB
S	1	2
S+1	3	4
S+2	5	6
S+3	7	8
..		

**Замечание** MSB - старшие байты, LSB - младшие байты

Когда цифра 0 в слове C = 0, биты источника, показанные в таблице, будут переданы в порядке: 12345678..

Когда цифра 0 в слове C = 1, биты источника, показанные в таблице, будут переданы в порядке: 21436587..

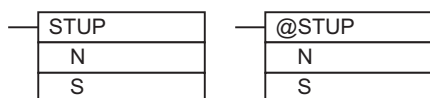
**Замечание** Когда заданы коды начала и конца, общая длина блока данных должна быть максимум 256 байт, включая коды начала и конца.

**Флаги**

- ER:** К указанному порту не подключено ответное устройство.  
Ошибка в задании установочных параметров связи или оперативных параметров.
- Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- Слова источника (S.. S+(N\_2)-1) вышли за пределы области данных.
- 26405:** Флаг разрешения связи по порту RS-232C
- 26413:** Флаг разрешения связи по периферийному порту
- 26705:** Флаг разрешения связи по блоку HOST LINK #0
- 26713:** Флаг разрешения связи по блоку HOST LINK #1

**5.28.3 STUP - Изменение установочных параметров RS-232C**

**Обозначение на ПКС**



Контроллер	Номер инструкции	Операнды		
		№	N	S
			определитель порта	первое слово блока данных
C200H-CPU-E	-	IR 000, IR 001, IR 002	IR, SR, HR, AR, LR, TC, DM, #	
C200H-CPU-ZE	237	IR 000, IR 001, IR 002	IR, SR, HR, AR, LR, TC, DM, EM, #	

**Ограничения**

N должно быть IR 000, IR 001 или IR 002.

S и S+4 должны быть в одной области.

S можно задать #0000 для установки параметров порта RS-232C по умолчанию.

STUP нельзя выполнять для внутреннего порта RS-232C. если секция 2 переключателя DIP = ON.

STUP нельзя выполнять в подпрограмме прерываний.

**Описание**

Когда условие исполнения = 0, команда STUP не выполняется. Когда условие исполнения = 1, команда STUP изменяет установочные параметры порта на заданные в слове N.

N определяет, какая часть установочных параметров будет изменена.

N	Заданный порт
IR 000	Встроенный порт (параметры DM 6645.. DM 6649)
IR 001	Порт А панели связи (параметры DM 6555.. DM 6559)
IR 002	Порт В панели связи (параметры DM 6550.. DM 6554)

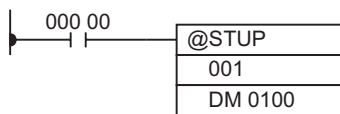
Если S - адрес слова, содержимое S.. S+4 копируется в 5 слов области параметров, которые содержат параметры для порта, заданного в N.

Если S - константа #0000, установочные параметры указанного порта устанавливаются по умолчанию.

S	Функция
Адрес слова	Содержание S.. S+4 копируется в область установочных параметров ПК, которые содержат параметры для порта, заданного в N.
Константа #0000	Установочные параметры указанного порта устанавливаются по умолчанию.

**Пример применения**

В данном примере показана программа, которая передает содержимое DM 0100.. DM 0104 в область установочных параметров порта А панели связи (DM 6555.. DM 6559).



Адрес	Инструкция	Операнд
00000	LD	000 00
00001	@STUP	01 DM 0100

Параметры передаются, как показано ниже. Флаг изменения параметров RS-232C (SR 27504) установится в 0 после завершения передачи.

DM 0100	1001	DM 6555	1001
DM 0101	0803	DM 6556	0803
DM 0102	0000	DM 6557	0000
DM 0103	2000	DM 6558	2000
DM 0104	0000	DM 6558	0000

В таблице показана функция передаваемых параметров.

Слово	Содержание	Функция
DM 0100	1001	Разрешает взять задание из DM 0101, и задет режим связи RS-232C.
DM 0101	0803	Определяет следующие параметры связи: 9 600 бод, 1 стартовый бит, 8 бит данных, 1 стоповый бит, нет контроля четности.
DM 0102	0000	Нет задержки передачи (0 мс)
DM 0103	2000	Разрешает код окончания (CR, LF).
DM 0104	0000	—

**Флаги**

**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

Определитель порта (N) не IR 000, IR 001 или IR 002.

Задан порт A, секция 2 переключателя DIP = ON.

Установочные параметры защищены от записи (секция 1 переключателя DIP = ON).

Слова источника превысили область данных.

Команда выполнялась из подпрограммы прерываний.

**5.28.4 PMCR - Функция поддержки протоколов****Обозначение на ПКС**

PMCR	@PMCR
C	C
S	S
D	D

Контроллер	Номер инструкции Nr	Операнды		
		S	C	D
		слово состояния	первое входное слово	первое выходное слово
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM
C200H□-CPU□□-ZE	260	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM

**Ограничения**

C должно быть двоично-десятичным числом от #0001 до #2999.

DM 6144.. DM 6655 нельзя использовать для D

**Описание**

Когда условие исполнения = 0, команда PMCR не выполняется. Когда условие исполнения = 1, команда PMCR вызывает и выполняет заданный протокол связи, который был зарегистрирован в панели связи, установленной ПК.

Когда для S и D задан не DM, сообщение передачи/приема для протокола связи, зарегистрированного в панели связи ПК, должно быть настроено на чтение или запись данных типа слово. Когда нет необходимости задавать слово в качестве первого слова выхода, используйте константу.

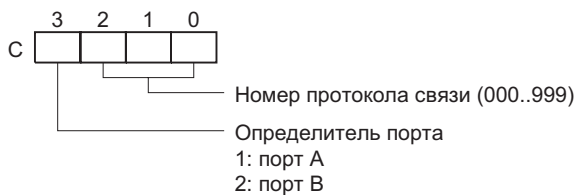
Когда протокол связи не требует входного слова, все равно задайте адрес слова. Данные не будут загружены в указанное слово и содержимое слово сохранится. Когда последовательность связи требует слов входа, задайте адрес слов, которые не используются для других целей в данной программы.

Входное и выходное слова (S и D) можно также задать в протоколе связи, зарегистрированном в панели связи.

**Замечание** Подробности о панелях связи см. инструкцию по работе с панелями связи и подробности о протоколе связи см. инструкцию по работе с протоколом.

### Слово управления

Первая цифра слова управления (1 или 2) задает порт панели связи, а последние три слова определяют протокол (000.. 999), как показано на диаграмме.



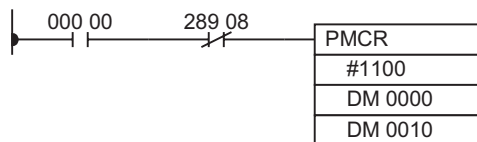
### Флаги

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- D не двоично-десятичное число или DM 6144..DM 6655 используется для D.
- При вызове команды уже работает еще одна команда PMCR.
- Определитель порта не 1 или 2.

### Пример

Когда IR 00000 = 1 и SR 28908 (флаг исполнения порт А панели связи) = 0, вызывается протокол связи 100 и данные передаются через порт А панели связи.

Посланные данные читаются из зоны слов, начинающихся в DM 0000 (первое слово выхода) и полученные данные сохраняются в словах, начинающихся в DM 0010 (первое слово входа).



Адрес	Инструкция	Операнд
00200	LD	000 00
00201	AND NOT	289 098
00202	PMCR	#1100
		DM 0000
		DM 0010

## 5.29 Дополнительные команды ввода/вывода

Дополнительные команды ввода/вывода позволяют проводить управление одной командой те операции, которые ранее были сложными; они включают внешние устройства ввода/вывода (цифровые переключатели, 7-сегментный индикатор и т. д.).

Есть 5 дополнительных команд ввода/вывода, показанные в следующей таблице. Перед применением всем им нужно присваивать функциональные коды.

Название	Мнемоника	Функция
7-SEGMENT DISPLAY OUTPUT	7SEG	Выдача двоично-десятичного числа на 7-сегментный индикатор
DIGITAL SWITCH INPUT	DSW2	Ввод данных с цифрового переключателя
HEXADECIMAL KEY INPUT	HKY	Ввод 16-ричного числа с клавиатуры 16 клавиш
TEN-KEY INPUT	TKY	Ввод десятичного числа с клавиатуры 10 клавиш
MATRIX INPUT	MTR	Ввод данных с матрицы 8 x 8

Хотя TKY служит только для упрощения программирования, другие дополнительные команды используются для сокращения времени цикла, сокращают использование специальных блоков ввода/вывода и уменьшают стоимость системы. За исключением TKY дополнительные команды можно использовать в программе только один раз и нельзя применить для блоков входов/выходов, установленных на ведомой раме для специальных блоков входов/выходов.

### 5.29.1 DSW - Ввод с цифрового переключателя

#### Обозначение на ПКС

DSW
IW
OW
R

Контроллер	Номер инструкции Nr	Операнды		
		IW входное слово	OW выходное слово	R первое слово результата
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	260	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, DM, EM

#### Описание

DSW служит для чтения значения, набранного на цифровом переключателе, подключенном к входным/выходным блокам. Когда условие исполнения = 0, команда DSW не выполняется. Когда условие исполнения = 1, команда DSW считывает 8 цифр, набранных на кодовом колесе, из IW и посылает результат в R.

8-разрядное число посылаются в R и R+1, старшие цифры в R+1.

DSW считывает 8-разрядное слово за 20 циклов, затем стартует опять и продолжает считывать данные.

Цифровой переключатель должен иметь 4 линии данных и один триггер-фиксатор для каждой считываемой цифры.

#### Предосторожности

Для обеспечения эффективности при каждом исполнении DSW должно выполняться обновление всех входов/выходов, которые использует команда DSW. Команда обновления входов/выходов должна использоваться с DSW, если она используется в программе прерываний, чтобы обеспечить обновление точек входов/выходов. Пример такого типа программирования см. 5.28.5.

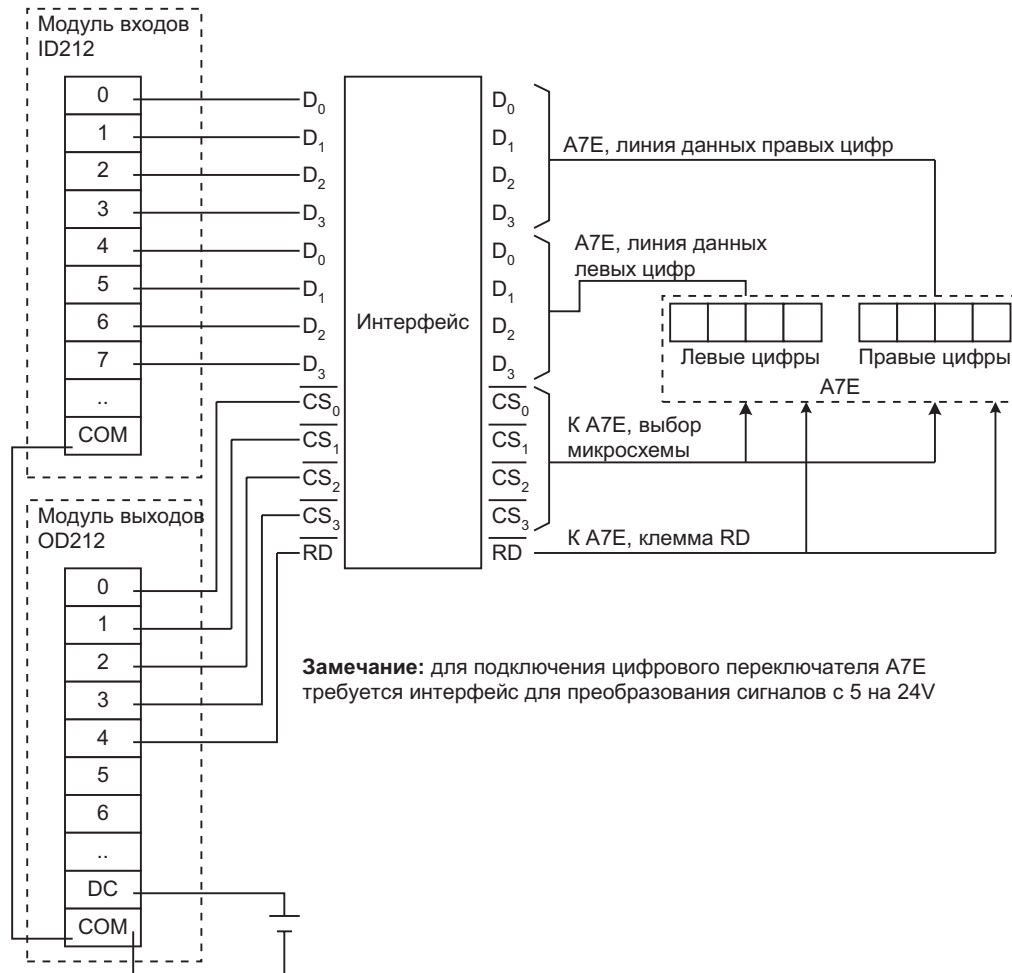
DSW будет исполняться с первого цикла после начала программы, включая перезапуск после пропадания питания.

Используйте команду DSW не более 2 раз в программе.

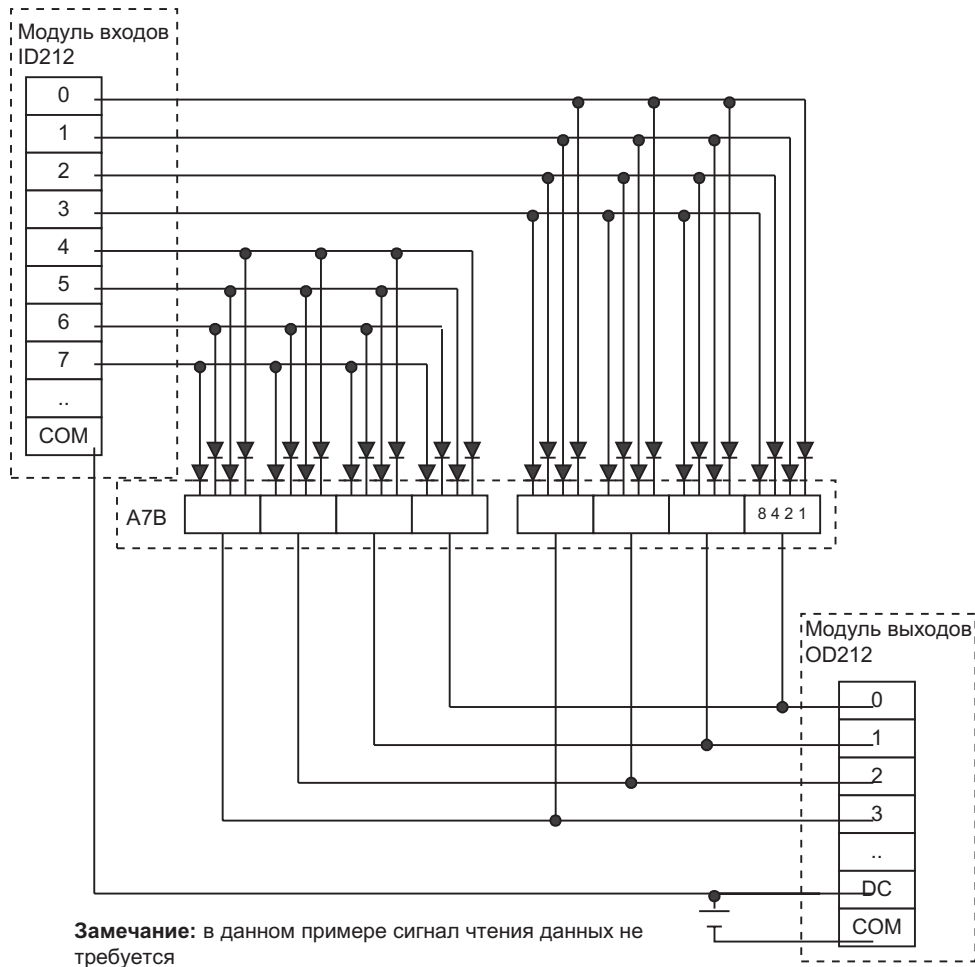
**Замечание** DSW нельзя использовать для блоков входов/выходов, установленных на ведомых рамах.

**Аппаратная часть**

При исполнении данной команды 8-разрядное двоично-десятичное число читается с цифрового переключателя. DSW использует 5 битов выхода и 8 битов входа. Подключите цифровой переключатель и блоки входа и выхода в соответствии со следующей схемой. Выходная точка 5 будет = 1 после одного прохода чтения, но подключать его не обязательно, если нет особых задач.



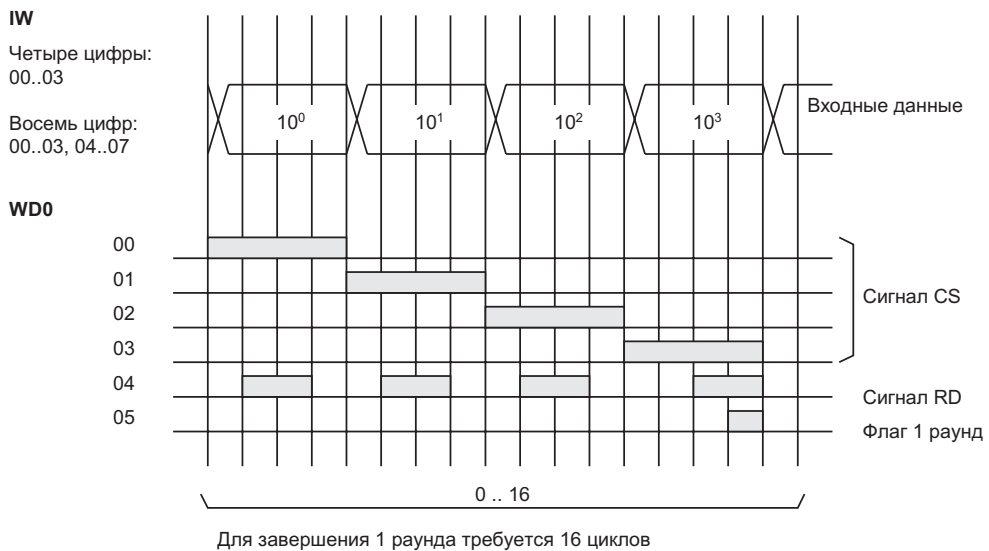
На схеме показано подключение Кодировочного колеса A7B.



Можно пользоваться входами как на ЦПУ, так и входами с блоков на 8 и более входов постоянного тока, выходы можно подключать к блокам транзисторных выходов на 8 выходов и более.

**Пример применения**

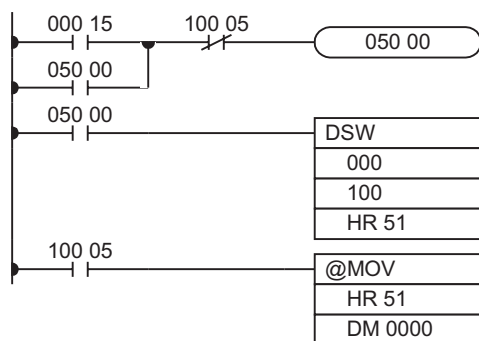
Если входное слово для подключения цифрового переключателя определено как слово IW, Слово выхода как OW, при выполнении программы операция будет происходить следующим образом:





**Пример применения**

В данном примере показана программа для чтения 8 цифр двоично-десятичного кода с цифрового переключателя. Цифровой переключатель подключен к входам IR 000 и выходам IR 100.



Когда IR 00015 = 1, IR 05000 будет самоподдерживаться в состоянии 1, пока флаг одного прохода (IR 10005) включится в 1 после завершения одного прохода чтения командой DSW.

Данные с цифрового переключателя загружаются командой DSW в HR 51.

Когда флаг одного прохода (IR 10005) = 1 после завершения чтения, число, сохраненное в HR 51 передается в DM 0000.

**Флаги**

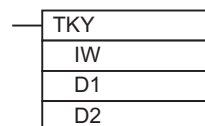
**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).

R и R+1 находятся в разных областях.

**SR 25410:** 1, когда идет выполнение DSW.

**5.29.2 ТКУ - Ввод с клавиатуры на 10 клавиш**

**Обозначение на ПКС**



Контроллер	Номер инструкции Nr	Операнды		
		IW	D1	D2
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	211	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, DM, EM

**Ограничения**

D1 и D1+1 должны находиться в одной области.

**Описание**

Когда условие исполнения = 0, команда ТКУ не выполняется. Когда условие исполнения = 1, команда ТКУ вводит данные с клавиатуры на 10 клавиш, подключенной к входу, заданному в IW. Данные можно вводить двумя способами:

ТКУ можно использовать в программе несколько раз, изменив входное слово IW.

- 1, 2, 3,... 1. В D1 и D1+1 создается 8-разрядный сдвиговый регистр. Когда нажата клавиша на клавиатуре, соответствующая двоично-десятичная цифра вдвигается в младший разряд D1. Другие цифры D1, D1+1 сдвигаются влево и самая старшая цифра D1+1 теряется.
2. Первые 10 бит слова D2 определяют вход клавиши. Когда одна из клавиш (0.. 9) нажата, соответствующий бит в D2 (00..09) устанавливается в 1.

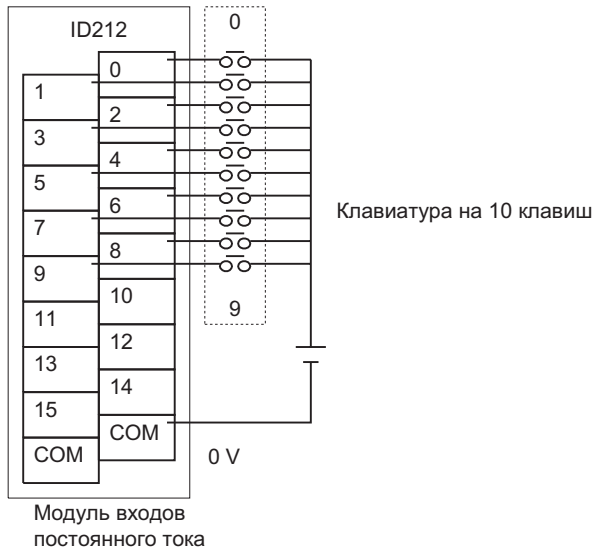
**Замечание**

1. Когда нажата клавиша, ввод с других клавиш не происходит.

2. Если вводится более 8 цифр, они будут удаляться, начиная с самой левой.
3. Неиспользованные биты входа можно использовать как обычные входные биты.

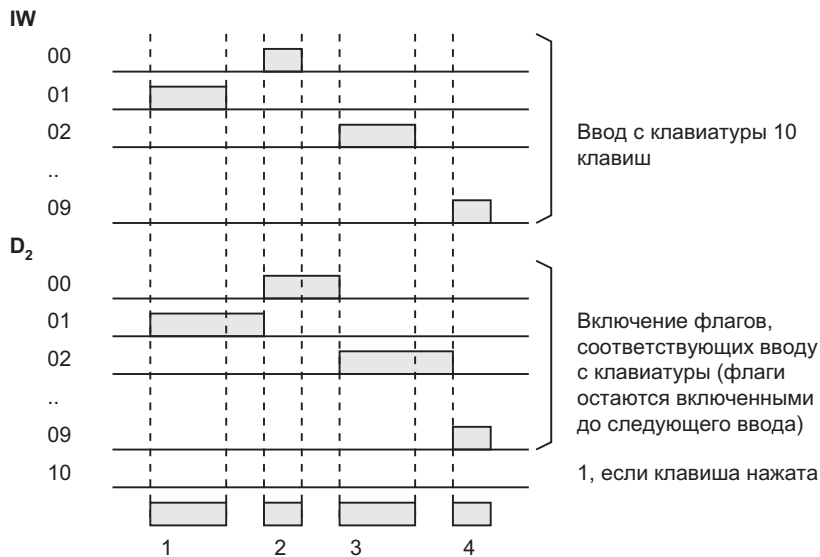
**Аппаратная часть**

Данная команда вводит 8 двоично-десятичных цифр с клавиатуры на 10 клавиш и использует 10 входных точек. Приготовьте клавиатуру на 10 клавиш и подключите их так, чтобы клавиши 0 - 9 являлись входами 0 - 9, как показано на схеме. Можно пользоваться и входами на ЦПУ и входами с блоков на 16 и более входов постоянного тока.



**Применение команды**

Если для IW задано слово входа с клавиатурой 10 клавиш, при выполнении программы операция будет происходить следующим образом:



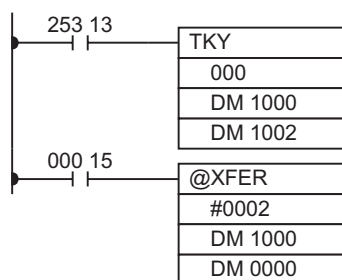
	D <sub>1</sub> +1				D <sub>1</sub>				
Перед исполнением	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	1	Нажата клавиша '1'
2	0	0	0	0	0	0	1	0	Нажата клавиша '0'
3	0	0	0	0	0	1	0	2	Нажата клавиша '2'
4	0	0	0	0	1	0	2	9	Нажата клавиша '9'

**Флаги**

**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).  
D1 и D1+1 находятся в разных областях.

**Пример**

В данном примере показана программа для ввода с клавиатуры на 10 клавиш, которая подключена к входу IR 000.



Информация, введенная в IR 000 с помощью ТКУ, преобразуется в двоично-десятичный вид и хранится в DM 1000 - DM 1001. Информация о клавишах хранится в DM 1002.

IR 00015 используется в качестве клавиши "ВВОД", и когда IR 00015 = 1, данные, хранящиеся в DM 1000 и DM 1001 будут переписаны в DM 0000 и DM 0001.

**5.29.3 НКУ - Ввод с клавиатуры на 16 клавиш**

**Обозначение на РКС**

НКУ
IW
OW
R

Номер инструкции		Операнды		
Контроллер	№	IW	OW	D
		входное слово		
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	212	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, DM, EM

**Ограничения**

D и D+2 должны находиться в одной области.

**Описание**

Когда условие исполнения = 0, команда НКУ не выполняется. Когда условие исполнения = 1, команда НКУ вводит данные с клавиатуры на 16 клавиш, подключенной к входу, заданному в IW. Данные можно вводить двумя способами:

- 1, 2, 3,... 1. В D и D+1 создается 8-разрядный сдвиговый регистр. Когда нажата клавиша на клавиатуре, соответствующая 16-ричная цифра вдвигается в младший разряд D. Другие цифры D, D+1 сдвигаются влево и самая старшая цифра D+1 теряется.
2. Биты слова D+2 и бит 4 OW указывают на ввод клавиши. Когда одна из клавиш (0.. F) нажата, соответствующий бит в D+2 (00..15) и бит 4 OW устанавливается в 1.

- Замечание**
1. Когда нажата одна из клавиш, ввод с других клавиш невозможен.
  2. Входные и выходные биты, не используемые здесь, можно использовать в качестве входных и выходных битов.

В данной команде одна клавиша читается от 4 до 13 циклов. Несколько циклов требуется в связи с тем, что клавиши в состоянии 1 можно определить после включения в 1 выходов для их тестирования.

Устройство 16-клавиш должно подключаться в виде матрицы 4x4.

**Предосторожности**

Для обеспечения эффективности при каждом исполнении НКУ должно выполняться обновление всех входов/выходов, которые использует команда НКУ. Команда обновления входов/выходов должна использоваться с НКУ, если она используется в программе прерываний, чтобы обеспечить обновление точек входов/выходов. Пример такого типа программирования см. 5.28.5.

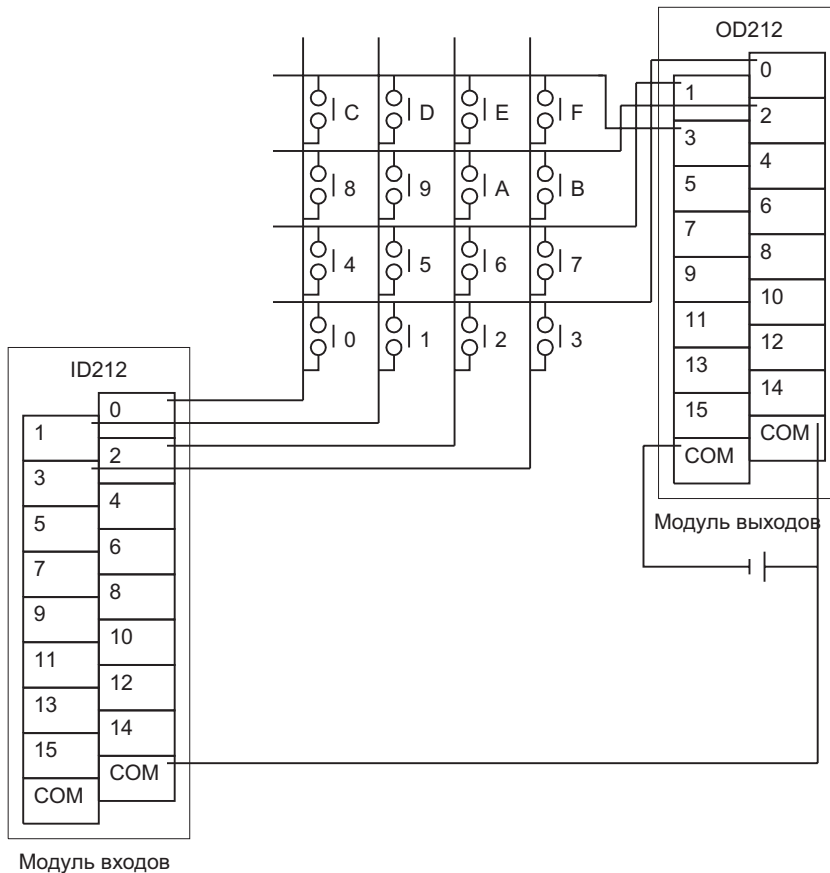
НКУ будет исполняться с первого цикла после начала программы, включая перезапуск после пропадания питания.

Используйте команду НКУ не более 2 раз в программе.

НКУ нельзя использовать для блоков входов/выходов, установленных на ведомых рамах.

**Аппаратная часть**

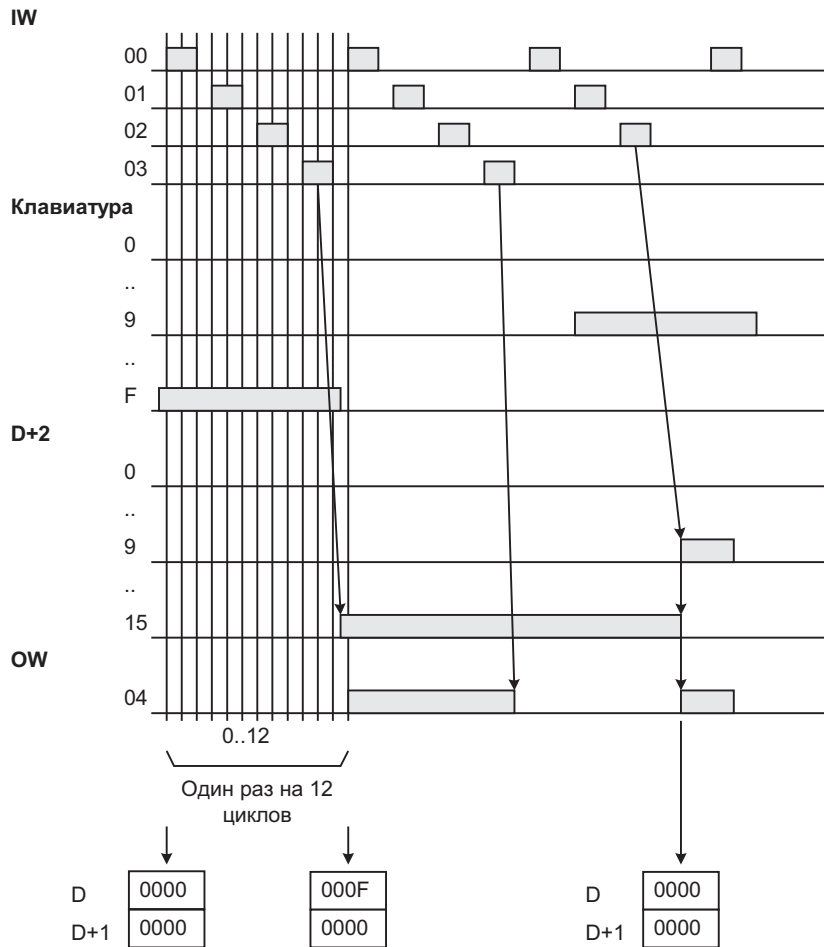
Данная команда вводит 8 цифр в 16-ричном виде с клавиатуры на 16 клавиш. Она использует 5 выходных битов и 4 входных бита. Приготовьте клавиатуру 16 клавиш и подключите клавиши 0..F в соответствии со схемой, к входам 0..3 и выходам 0..3. Выход 4 будет = 1 при нажатии любой клавиши, но подключать его не требуется, если нет специальных задач.



Входы, подключенные к входным клеммам, должны быть на 8 и более входов постоянного тока, выходы подключенные к входным клеммам, должны быть с блока транзисторных выходов на 8 и более входных точек.

**Применение команды**

Если для IW задано слово ввода с клавиатуры 16 клавиш, и выходное слово задано в OW, при выполнении программы операция будет происходить следующим образом:



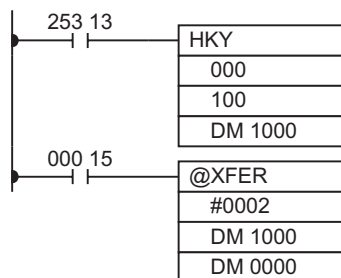
**Флаги**

**ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).  
D и D+2 находятся в разных областях.

**SR 25408:** 1, когда идет выполнение НКУ.

**Пример**

В данном примере показана программа для ввода с клавиатуры на 16 клавиш, которая подключена к входам IR 000 и выходам IR 100.

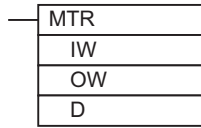


Информация, введенная в IR 000 с помощью НКУ, преобразуется в 16-ричный вид и хранится в DM 1000 - DM 1001.

IR 00015 используется в качестве клавиши "ВВОД", и когда IR 00015 = 1, данные, хранящиеся в DM 1000 и DM 1001 будут переписаны в DM 0000 и DM 0001.

### 5.29.4 MTR - Ввод матрицы

#### Обозначение на ПКС



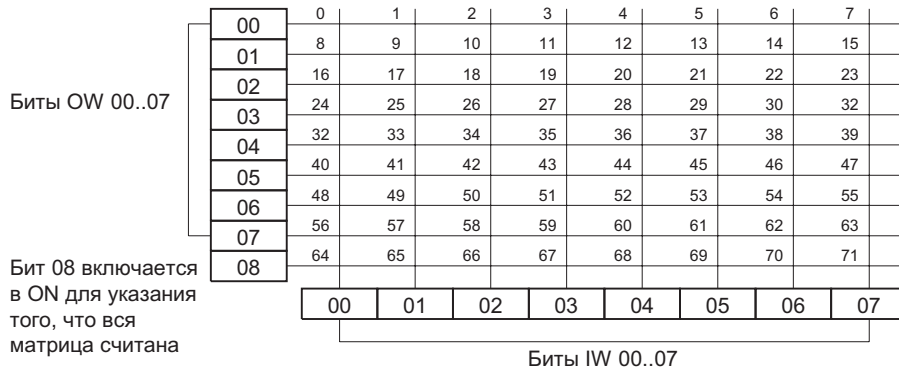
Контроллер	Номер инструкции №	Операнды		
		IW входное слово	OW выходное слово	D
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM	IR, SR, HR, AR, LR, DM
C200H□-CPU□□-ZE	213	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, DM, EM	IR, SR, HR, AR, LR, DM, EM

#### Ограничения

D1 и D1+3 должны находиться в одной области.

#### Описание

Когда условие исполнения = 0, команда MTR не выполняется. Когда условие исполнения = 1, команда MTR вводит данные с матрицы 8x8 и записывает их в D..D+3. Данные для всех 64 точек матрицы будут записаны, даже если подключено менее 64 клавиш.



Сигнал выбора выдается в OW биты 00..07 последовательно за 3 цикла. Только один выходной бит будет включен за раз. Бит 08 OW включится в 1 на 3 цикла после 07 для того, чтобы показать, когда каждый прогон чтения матрицы завершился.

Когда нажата одна из 64 клавиш, будет получен вход на один из входных битов. Нажатая клавиша определяется путем сравнения выходного бита, на который выдается сигнал и входного бита, на котором получен сигнал.

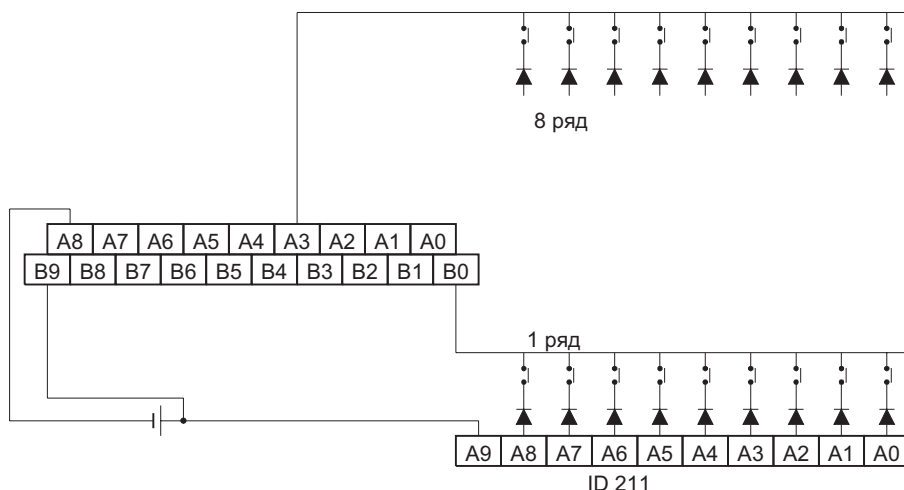
Когда обнаружено нажатие клавиши, соответствующий бит в D..D+3 включается в 1. В следующей таблице показано соответствие между клавишами и битами в D.. D+3.

Слово	Биты	Соответствующие клавиши
D	00..15	0..15
D+1	00..15	16..31
D+2	00..15	32..47
D+3	00..15	48..63

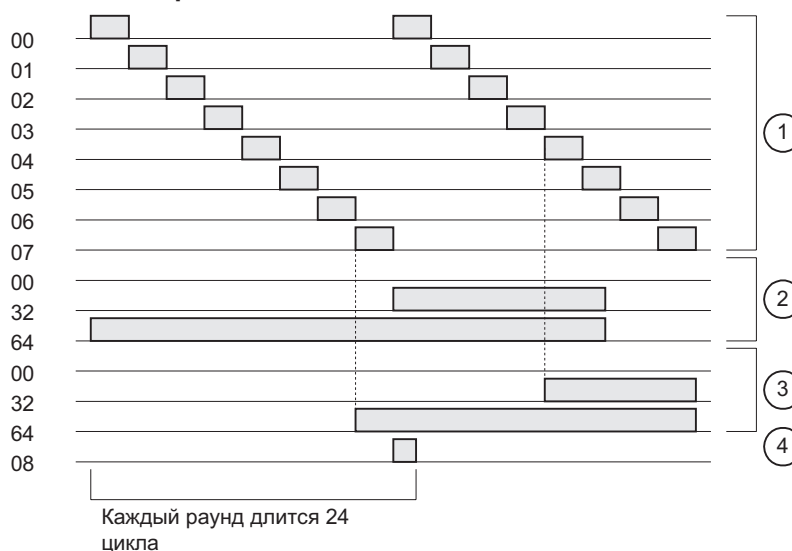
#### Аппаратная часть

Данная команда вводит до 64 сигналов с матрицы 8x8, используя 8 входных точек и 8 выходных точек. Можно использовать любую матрицу. Входы должны быть подключены через блок входов постоянного тока с 8 и более точками, а выходы должны быть подключены через блок транзисторных выходов с 8 и более точками. Основная схема подключения и временная диаграмма показана на рисунке.

## Подключение



## Временная диаграмма



1. Сигнал выбора
2. Состояние матрицы
3. Биты, указывающие на состояние матрицы
4. Флаг одного раунда (бит 08 выходного слова)

## Предосторожности

64 клавиши можно разделить на 8 рядов (включая ряд для OW бит 08), которые сканируются последовательно. Поскольку каждый ряд сканируется за 3 цикла, может получиться задержка до 24 циклов перед тем, как данный ряд будет сканирован для входов.

Для обеспечения эффективности при каждом исполнении MTR должно выполняться обновление всех входов/выходов, которые использует команда MTR. Команда обновления входов/выходов должна использоваться с MTR если она используется в программе прерываний, чтобы обеспечить обновление точек входов/выходов.

MTR будет исполняться с первого цикла после начала программы, включая перезапуск после пропадания питания.

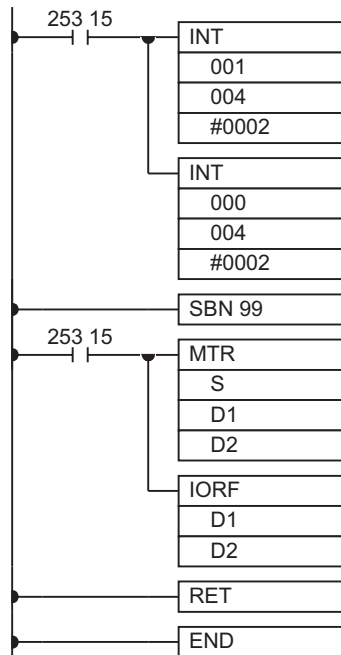
SR 25403, который включается в 1 во время исполнения MTR, сбрасывается в секциях INTERLOCK и MTR не исполняется в секциях INTERLOCK.

Используйте команду MTR не более 2 раз в программе.

MTR нельзя использовать для блоков входов/выходов, установленных на ведомых рамах.

**Пример**

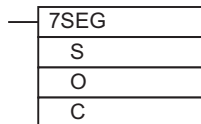
В следующем примере показано программирование MTR в прерываниях по расписанию, где IOFR запрограммирована для обеспечения обновления слов входов/выходов, используемых командой MTR, при каждом исполнении MTR.



**Флаги**

- ER:** Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).
- SR 25403:** 1, когда идет выполнение MTR.

**5.29.5 7SEG - Вывод на 7-сегментный индикатор**



Контроллер	Номер инструкции Nг	Операнды		
		S первое слово источника	O выходное слово	C слово состояния
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, DM	# (000..007)
C200H□-CPU□□-ZE	214	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, DM, EM	# (000..007)

**Ограничения**

- S и S+1 должны быть в одной области данных.
- DM 0000..DM 6143 можно использовать для O
- Значения C могут быть только 000..007.

**Описание**

Когда условие исполнения = 0, команда 7SEG не выполняется. Когда условие исполнения = 1, команда 7SEG читает данные источника (либо 4, либо 8 цифр), преобразует их в данные 7-сегментного индикатора и выдает эти данные на 7-сегментный индикатор, подключенный к выходу, заданному в O.

Значение C задает количество цифр источника и логику выходного блока ПК и входного регистра индикатора, как показано в следующей таблице.



## 5.29 Дополнительные команды ввода/вывода

Данные источника	Логика входов индикатора	Логика входов триггера индикатора	C
4 цифры (S)	Аналогична блоку выходов ПК	Аналогична блоку выходов ПК	000
		Отлична от блоков выходов ПК	001
	Отлична от блоков выходов ПК	Аналогична блоку выходов ПК	002
		Отлична от блоков выходов ПК	003
8 цифр (S, S+1)	Аналогична блоку выходов ПК	Аналогична блоку выходов ПК	004
		Отлична от блоков выходов ПК	005
	Отлична от блоков выходов ПК	Аналогична блоку выходов ПК	006
		Отлична от блоков выходов ПК	007

Если данные источника - 8 цифр, они хранятся в S и S+1, старшие цифры в S+1.

Если данные источника - 4 цифры, они хранятся в S.

7SEG индицирует 4-разрядные или 8-разрядные данные за 12 циклов, затем стартует опять и продолжает индицировать данные.

7-сегментный индикатор должен иметь 4 линии данных и один триггер-фиксатор для каждой индицируемой цифры.

- Замечание**
1. При разработке системы учитывайте время цикла и характеристики 7-сегментного индикатора.
  2. Неиспользованные выходные биты можно использовать в качестве обычных выходных бит.

### Предосторожности

Для обеспечения эффективности при каждом исполнении 7SEG должно выполняться обновление всех входов/выходов, которые использует команда 7SEG. Команда обновления должна использоваться с 7SEG, если она используется в программе прерываний, чтобы обеспечить обновление точек входов/выходов. Пример такого типа программирования см. 5.28.5.

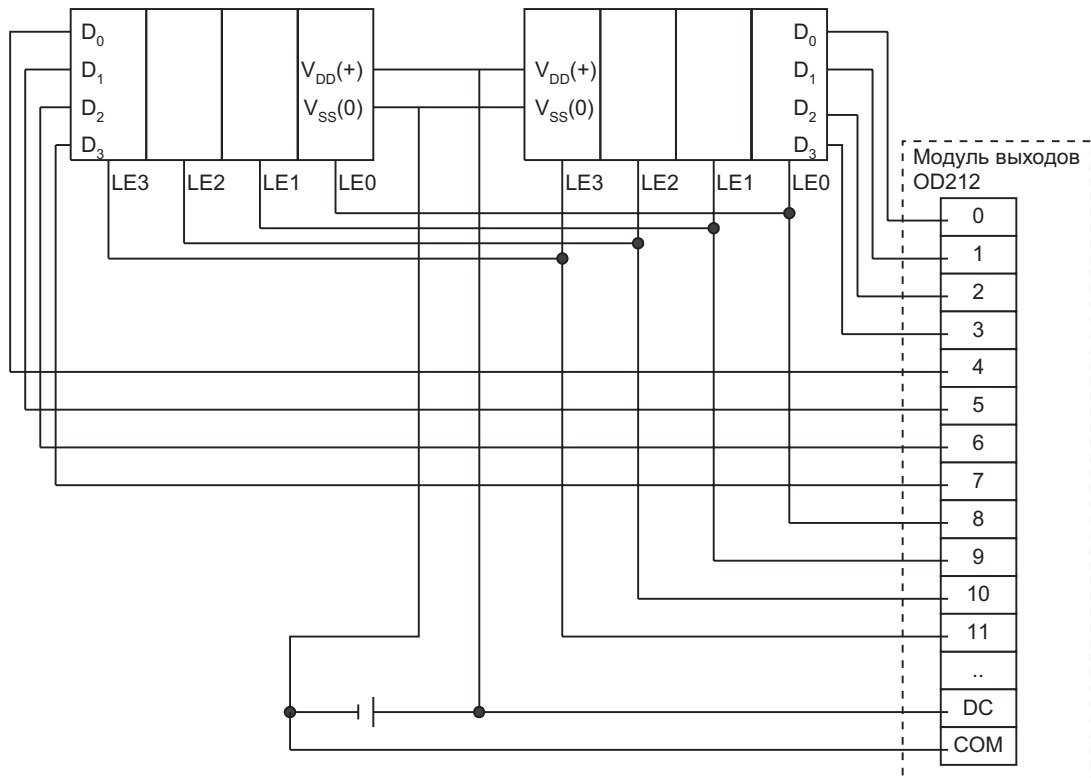
7SEG будет исполняться с первого цикла после начала программы, включая перезапуск после пропадания питания.

Используйте команду 7SEG не более 2 раз в программе.

7SEG нельзя использовать для блоков входов/выходов, установленных на ведомых рамах.

### Аппаратная часть

Данная команда посылает число на 7-сегментный индикатор. Она использует либо 8 битов выхода для 4 цифр, либо 12 выходных битов для 8 цифр. 7-сегментный индикатор подключается к блоку выходов в соответствии со схемой. Для вывода 4 цифр, выходы данных (D0 - D3) подключаются к выходам 0 - 3, выходы триггера-фиксатора (CS0 - CS3) подключаются к выходам 4..7. Выход 12 (для вывода 8 цифр) или 8 (для вывода 4 цифр) = 1, когда высвечивается 1 блок данных, но подключать их не обязательно, если не требуется решаемой задачей.



Выходы должны подключаться к блокам транзисторных выходов на 8 выходов и более для вывода 4 цифр или на 16 выходов для вывода 8 цифр. Можно использовать блоки базовые выходы, специальные блоки входов/выходов, или блоки выходов высокой плотности.

- Замечание**
1. Блоки выходов обычно используют отрицательную логику. (Только выходы PNP используют положительную логику).
  2. 7-сегментный индикатор может требовать либо позитивной, либо негативной логики в зависимости от модели.
  3. 7-сегментный индикатор должен иметь 4 линии данных и один триггер-фиксатор для каждой индицируемой цифры.

#### Применение команды

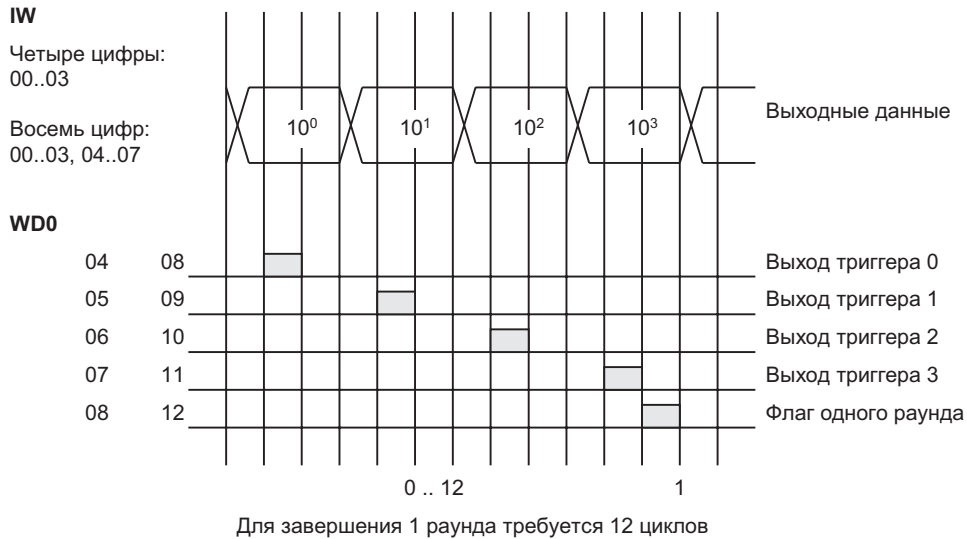
Если первое слово, содержащее данные для индикации, определено как S, Слово выхода как O, и константа из таблицы как C, при выполнении программы операция будет происходить следующим образом. Если выводятся только 4 цифры, используется только слово S.

#### Формат хранения данных

Левые 4 цифры	Правые 4 цифры
S+1	S

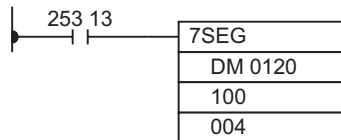
#### Синхронизация

Синхронизация выдачи данных приведена показана далее. Состояние выхода (данные и логика триггера зависят от C)



**Пример применения**

В данном примере показана программа для вывода 8 цифр на 7-сегментный светодиодный индикатор, который подключен к выходному слову IR 100. Предполагается, что выходной блок с негативной логикой, 7-сегментный индикатор также с негативной логикой и для данных и для сигналов триггера.



8 двоично-десятичных цифр DM 0120 (4 младшие цифры) и DM 0121 (4 старшие цифры) постоянно выводятся командой 7SEG. При изменении DM 0120 и DM 0121 индикация также изменяется.

**Флаги**

**ER:** S и S+1 находятся в разных областях (При индикации 8-разрядных данных). Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).  
Ошибка в задании операнда.

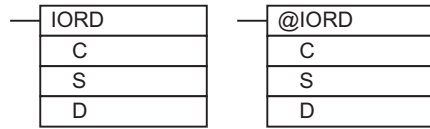
**SR 25409:** 1, когда идет выполнение 7SEG.

## 5.30 Команды для работы со Специальными Модулями

Команды блоков специальных входов/выходов используются для передачи данных из и в память определенного блока специальных входов/выходов.

### 5.30.1 IORD - Чтение Специальных модулей

#### Обозначение на ПКС



Контроллер	Номер инструкции №	Операнды		
		C	S	D
C200H□-CPU□□-E	-	слово состояния		
C200H□-CPU□□-ZE	222	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM, EM

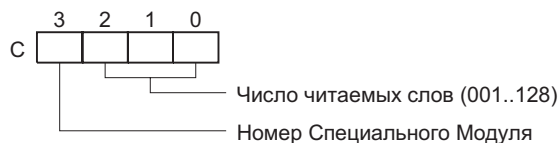
#### Ограничения

Можно задавать только блоки специальных входов/выходов, установленные на раме ЦПУ или рамах расширенных входов/выходов.

Последние 3 цифры S должны быть двоично-десятичным числом (001.. 128).

#### Описание

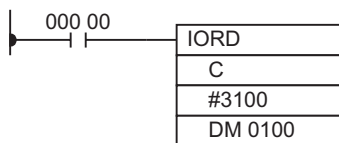
Когда условие исполнения = 0, команда IORD не исполняется. Когда условие исполнения = 1, IORD передает данные из памяти заданных блоков специальных входов/выходов в слова области D. Информация об источнике - это номер узла блока специальных входов/выходов и количество читаемых слов, как показано на диаграмме.



Управляющий код C зависит от типа блока специальных входов/выходов. Подробности см. соответствующую инструкцию.

#### Пример

Когда IR 00000 переходит из 0 в 1, следующая команда переносит 100 слов из области памяти блока специальных входов/выходов номер 3 в DM 0100.. DM 0199.



Адрес	Инструкция	Операнд
00200	LD	000 00
00201	@IORD	C
		#3100
		DM 0100

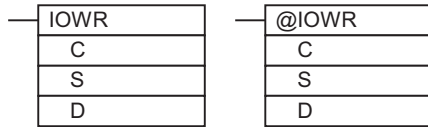
#### Флаги

- ER:** Последние 3 цифры S (определителя количества слов) не двоично-десятичное число или не в диапазоне 001.. 128.  
Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).  
Номер блока источника не 0.. F или он смонтирован не ведомой рамой.  
Полученные данные выходят за допустимые границы.

**EQ:** 1, когда данные считались успешно, в противном случае = 0.

### 5.30.2 IOWR - Запись в Специальные модули

#### Обозначение на ПКС



Контроллер	Номер инструкции №	Операнды		
		C слово состояния	S	D
C200H□-CPU□□-E	-	IR, SR, HR, AR, LR, TC, DM, #	IR, SR, HR, AR, LR, TC, DM	IR, SR, HR, AR, LR, TC, DM, #
C200H□-CPU□□-ZE	223	IR, SR, HR, AR, LR, TC, DM, EM, #	IR, SR, HR, AR, LR, TC, DM, EM	IR, SR, HR, AR, LR, TC, DM, EM, #

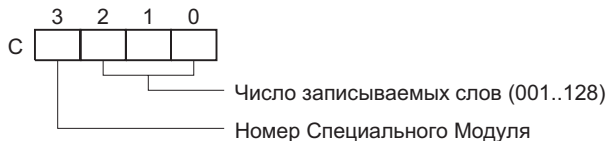
#### Ограничения

Можно задавать только блоки специальных входов/выходов, установленные на раме ЦПУ или рамках расширенных входов/выходов.

Последние 3 цифры D должны быть двоично-десятичным числом (001.. 128).

#### Описание

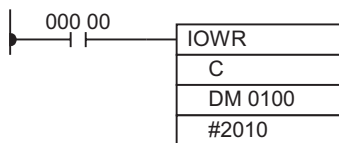
Когда условие исполнения = 0, команда IOWR не исполняется. Когда условие исполнения = 1, IOWR передает данные из блока слов, начинающихся с D в память заданных блоков специальных входов/выходов. Информация о приемнике - это номер узла блока специальных входов/выходов и количество записываемых слов, как показано на диаграмме.



Управляющий код C зависит от типа блока специальных входов/выходов. Подробности см. соответствующую инструкцию.

#### Пример

Когда IR 00000 переключается из 0 в 1, следующая команда переносит содержание 10 слов DM 0100.. DM 0109 в область памяти блока номер 2 специальных входов/выходов.



Адрес	Инструкция	Операнд
00200	LD	000 00
00201	@IOWR	C
		DM 0100
		#2010

#### Флаги

- ER:** Последние 3 цифры D (определителя количества слов) не двоично-десятичное число или не в диапазоне 001.. 128.  
 Косвенно адресуемое слово DM не существует. (Содержимое слова \*DM не в двоично-десятичном виде, или выходит за границу области DM).  
 Номер блока приемника не 0.. F или он смонтирован не ведомой рамой.  
 Команда не завершилась нормально.

**EQ:** 1, когда данные записались успешно, в противном случае = 0.



# Приложение А

## Стандартные модели

### Панель ЦПУ

Наименование	Характеристики				Номер модели
	UM, Кслов	DM, Кслов	Число входов/ выходов	RS-232C	
ЦПУ (На всех моделях есть функция часов и платоместа для связи, исключая CPU11-E)	3.2	4	640	Нет	C200HE-CPU11-E
					C200HE-CPU11-ZE
	7.2	6	880	Нет	C200HE-CPU32-E
					C200HE-CPU32-ZE
				Да	C200HE-CPU42-E
					C200HE-CPU42-ZE
	15.2	12	880	Нет	C200HG-CPU33-E
					C200HG-CPU33-ZE
				Да	C200HG-CPU43-E
					C200HG-CPU43-ZE
			1 184	Нет	C200HG-CPU53-E
					C200HG-CPU53-ZE
				Да	C200HG-CPU63-E
					C200HG-CPU63-ZE
	31.2	24	880	Нет	C200HX-CPU34-E
					C200HX-CPU34-ZE
				Да	C200HX-CPU44-E
					C200HX-CPU44-ZE
1 184			Нет	C200HX-CPU54-E	
				C200HX-CPU54-ZE	
			Да	C200HX-CPU64-E	
				C200HX-CPU64-ZE	
63.2	48			C200HX-CPU65-ZE	
	96			C200HX-CPU85-ZE	
Модули питания	100..120/200..240 В пост. тока				C200HW-PA204
	100..120/200..240 В пост. тока (с сервисным выходом 0.8 A 24 В пост. тока)				C200HW-PA204S
	100..120/200..240 В пост. тока (с релейным выходом 2 A 250VAC/24VDC для индикации режима RUN )				C200HW-PA204R
	24 В пост. тока				C200HW-PD24
Базовые панели ЦПУ	3 платоместа				C200HW-BC031
	5 платомест				C200HW-BC051
	8 платомест				C200HW-BC081
	10 платомест				C200HW-BC101
Панели связи	Порт связи для Модулей связи SYSMAC LINK и SYSMAC NET				C200HW-COM01
	Порт RS-232C				C200HW-COM02-V1
	Порт RS-422/485				C200HW-COM03-V1
	Порт связи для Модуля связи SYSMAC LINK и SYSMAC NET и функции PROTOCOL MACRO				C200HW-COM04-EV1

Наименование	Характеристики		Номер модели
	2 порта RS-232C и функция PROTOCOL MACRO		C200HW-COM05-EV1
	Порт RS-422/485, порт RS-232C и функция PROTOCOL MACRO		C200HW-COM06-EV1
Кассеты памяти	EEPROM	4 К слов	C200HW-ME04K
		8 К слов	C200HW-ME08K
		16 К слов	C200HW-ME16K
		32 К слов	C200HW-ME32K
		64 К слов	C200HW-ME64K
	EPROM	16 К слов/ 32 К слов	C200HS-MP16K
		Эквивалент 27256, 150 нс, 12.5 В	ROM-JD-B
	Эквивалент 27512, 150 нс, 12.5 В	ROM-KD-B	

**Панели расширения**

Наименование	Характеристика		Номер модели
Модули питания	100..120/200..240 В пост. тока		C200HW-PA204
	100..120/200..240 В пост. тока (С выходами на 24 В пост. тока)		C200HW-PA204S
	24 В пост. тока		C200HW-PD24
Базовые панели	3 платоместа		C200HW-BI031
	5 платомест		C200HW-BI051
	8 платомест		C200HW-BI081
	10 платомест		C200HW-BI101
Соединительные кабели	30 см	Общая длина соединительных кабелей входов/выходов, используемых в сети, не должна превышать 12 м.	C200H-CN311
	70 см		C200H-CN 711
	2 м		C200H-CN221
	5 м		C200H-CN521
	10 м		C200H-CN131

**Ведомые панели**

Наименование	Характеристика		Номер модели
Ведомые панели	Ведомые Модули		
	100..120/200..240 В пер. тока (переключаемые)	APF/PCF	C200H-RT001-P
	24 В пост. тока		C200H-RT002-P
	100..120/200..240 В пер. тока (переключаемые)	Проводные	C200H-RT201
	24 В пост. тока		C200H-RT202
	Задние панели		
	3 платоместа		C200H-BC031-V2
	5 платомест		C200H-BC051-V2
	8 платомест		C200H-BC081-V2
	10 платомест		C200H-BC101-V2
Модули входов/выходов	Входы	Либо 12, либо 24 В пост. тока	G71-IC16
	Выходы		G71-OD16
Терминалы	Либо 100, либо 200 В пер. тока		G7TC-IA16
	Либо 12, либо 24 В пост. тока		G7TC-ID16
	Либо 12, либо 24 В пост. тока		G7TC-OC16



## Модули входов/выходов

Наименование	Характеристика		Номер модели
Модули входа	Модули входа пер. тока		
	8 точек	100..120 В пер. тока	C200H-IA121
	16 точек	100..120 В пер. тока	C200H-IA122/IA122V
	8 точек	200..240 В пер. тока	C200H-IA221
	16 точек	200..240 В пер. тока	C200H-IA222/IA222V
	Модули входа пост. тока		
	8 точек	12..24 В пост. тока	C200H-ID211
	16 точек	24 В пост. тока	C200H-ID212
	Модули входа пост. тока/пер. тока		
	8 точек	12..24 В пост. тока/пер. тока	C200H-IM211
	16 точек	24 В пост. тока/пер. тока	C200H-IM212
	Модуль входа прерывания (см. прим.)		
	8 точек	12..24 В пост. тока	C200HS-INT01
Модули выхода	Релейные Модули		
	8 точек	2А, 250 В пер.тока/24 В пост. тока (для активной нагрузки)	C200H-OC221
	12 точек	2А, 250 В пер.тока/24 В пост. тока (для активной нагрузки)	C200H-OC222
	12 точек	2А, 250 В пер.тока/24 В пост. тока (для активной нагрузки), независимые общие	C200H-OC222V
	5 точек	2А, 250 В пер.тока/24 В пост. тока (для активной нагрузки),	C200H-OC223
	8 точек	2А, 250 В пер.тока/24 В пост. тока (для активной нагрузки), независимые общие	C200H-OC224
	8 точек	2А, 250 В пер.тока/24 В пост. тока (для активной нагрузки), независимые общие	C200H-OC224V
	16 точек	2А, 250 В пер.тока/24 В пост. тока (для активной нагрузки), (см. прим.)	C200H-OC225
	16 точек	2А, 250 В пер.тока/24 В пост. тока (для активной нагрузки), (см. прим.)	C200H-OC226
	Модули симисторныхвыходов		
	8 точек	1А, 120 В пер.тока	C200H-OA121-E
	8 точек	1А, 250 В пер.тока	C200H-OA221
	12 точек	0.3А, 250 В пер.тока	C200H-OA222
	12 точек	0.3А, 250 В пер.тока	C200H-OA222V
	12 точек	1.2А, 250 В пер.тока	C200H-OA223
	12 точек	0.5А, 250 В пер.тока	C200H-OA224
	Модули транзисторныхвыходов		
	8 точек	1А, 12..48 В пост.тока	C200H-OD411
	12 точек	0.3 А, 24 В пост. тока	C200H-OD211
	16 точек	0.3 А, 24 В пост. тока (см. прим.)	C200H-OD212
	8 точек	2.1А, 24 В пост.тока	C200H-OD223
	8 точек	0.8 А, 24 В пост.тока; PNP, защита нагрузки от к/з	C200H-OD214

Наименование	Характеристика		Номер модели
	8 точек	0.3 А, 5..24 В пост.тока; PNP	C200H-OD216
	12 точек	0.3 А, 5..24 В пост.тока; PNP	C200H-OD217
	16 точек	1.0 А, 24 В пост.тока; PNP, защита нагрузки от к/з	C200H-OD21А
Модуль аналогового таймера	4 таймера	0.1..1 с/1..10 с/10..60 с/1 мин..10 мин (переключается)	C200H-TM001
	Разъем для переменного резистора	Разъем с проводом (2м) для 1 внешнего резистора	C4R-CN223
Модули интерфейса В7А	15 или 16 входов	Подключается к терминалам В7А. Стандартное время обмена.	C200H-B7А11
	16 входов	Подключается к терминалам В7А. Стандартное время обмена.	C200H-B7А01 (см. прим.)

**Замечание** Если Модуль входов с прерываниями смонтирован на панели расширения входов/выходов, функцию прерывания нельзя использовать и Модуль входов с прерываниями работает как обычный Модуль входов на 8 точек. Кроме того, Модули входов с прерываниями нельзя использовать на ведомой панели.

#### Модули входов/выходов группы-2

Наименование	Характеристика	Номер модели
Модули входа пост. тока	64 точки, 12 В пост. тока	C200H-ID111
	32 точки, 24 В пост. тока	C200H-ID216
	64 точки, 24 В пост. тока	C200H-ID217
Транзисторные Модули выхода	32 точки, 16 мА при 4.5 В.. 100 мА при 26.4 В	C200H-OD218
	64 точки, 16 мА при 4.5 В.. 100, мА при 26.4 В	C200H-OD219

#### Модули интерфейса В7А

Наименование	Характеристика		Номер модели
Модули интерфейса В7А	32 входа	Подключается к терминалам В7А. Время обмена стандартное или быстрое	C200H-B7А12
	32 выхода		C200H-B7А02
	16 входов, 16 выходов		C200H-B7А21
	32 входа, 32 выхода		C200H-B7А22

#### Специальные Модули

Наименование	Характеристика	№ модели
Модули входов/выходов высокой плотности (см. прим.)	Модули входов пост. тока	
	32 точки 5 В пост. тока; TTL; с высокоскоростным входом	C200H-ID501
	32 точки 24 В пост. тока; с высокоскоростным входом	C200H-ID215
	Модули транзисторных выходов	
	32 точки 0.1 А, 24 В пост. тока (используется как Модуль динамических выходов на 128 точек)	C200H-OD215
	32 точки 35 мА, 5 В пост. тока (TTL) (используется как Модуль динамических выходов на 128 точек)	C200H-OD218
	Модули входов пост. тока + транзисторных выходов	
	16 входов и 16 выходов, входы 24 В пост. тока; с высокоскоростным входом, выходы 0.1 А, 24 В пост. тока (используется как Модуль динамических выходов на 128 точек)	C200H-MD215

Наименование	Характеристика	№ модели
	16 входов и 16 выходов, входы 5 В пост. тока, TTL; с высокоскоростным входом, выходы 35 мА, 5 В пост. тока, выходы TTL (используется как Модуль динамических выходов на 128 точек)	C200H-MD501
	16 входов и 16 выходов, входы 12 В пост. тока, TTL; с высокоскоростным входом, выходы 12 В пост. тока, выходы TTL (используется как Модуль динамических выходов на 128 точек)	C200H-MD115
Модули аналоговых входов/выходов	Модули аналоговых входов	
	4 входа, 12 бит 4..20 мА, 1..5 В, 0..10 В	C200H-AD001
	8 входов, 12 бит, 4..20 мА, 1..5 В, 0..10 В, -10..10 В	C200H-AD002
	8 входов, 12 бит, 4..20 мА, 1..5 В, 0..10 В, -10..10 В	C200H-AD003
	Модули аналоговых выходов	
	2 выхода, 4..20 мА, 1..5 В, 0..10 В	C200H-DA001
	4 выхода, 4..20 мА, -10..10 В	C200H-DA002
	8 выходов, 1..5 В, 0..10 В, -10..10 В	C200H-DA003
	8 выходов, 4..20 мА	C200H-DA004
	Смешанный модуль	
2 входа 4..20 мА, 1..5 В, 0..10 В, -10..10 В 2 выхода 4..20 мА, 1..5 В, 0..10 В, -10..10 В	C200H-MAD01	
Модуль логики FUZZY	Прогрессирует с программного пакета поддержки FUZZY. До 8 входов и 4 выходов.	C200H-FZ001
	Пакет поддержки FUZZY	
	Поставляется на дискетах 3.5" или 5.25".	C500-SU981-E
Модули температурных датчиков	Термопара	
	4 входа, K(CA) или J(IC), переключается	C200H-TS001
	4 входа, K(CA) или L(Fe-CuNi) DIN стандарты	C200H-TS002
	Термометр на Pt резисторе	
	4 входа, Pt 100 Ом	C200H-TS101
	4 входа, Pt 100 Ом, DIN и 1989 JIS стандарты	C200H-TS102
Модули управления температурой	Термопара	
	Транзисторный выход	C200H-TC001
	Выход напряжения	C200H-TC002
	Токовый выход	C200H-TC003
	Термометр на Pt резисторе	
	Транзисторный выход	C200H-TC001
	Выход напряжения	C200H-TC002
Токовый выход	C200H-TC003	
Модули управления нагревом/охлаждением	Термопара	
	Транзисторный выход	C200H-TV001
	Выход напряжения	C200H-TV002
	Токовый выход	C200H-TV003
	Термометр на Pt резисторе	
	Транзисторный выход	C200H-TV001
	Выход напряжения	C200H-TV002
Токовый выход	C200H-TV003	
Модуль позиционирования по уставкам	Определяет углы поворота по резольверу и выдает 1 или 0 при заданных углах. Возможно максимум 48 уставок (16 внешних выходов и 32 внутренних выходов).	C200H-CP114

Наименование	Характеристика	№ модели
Пульт занесения данных	Служит для занесения и просмотра данных в Модули управления температуры, Модули позиционирования по уставкам, Модули PID-регулирования и Модули управления температурой нагревания/охлаждения.	C200H-DSC01
	Соединительный кабель	
	2 м	C200H-CN225
	4 м	C200H-CN425
Модули PID-регулирования	Транзисторный выход, вход 4..20 мА, 1.. 5 В, 0..5 В, 0..10 В (переключается)	C200H-PID01
	Выход напряжения, вход 4..20 мА, 1.. 5 В, 0..5 В, 0..10 В (переключается)	C200H-PID02
	Токовый выход, вход 4..20 мА, 1.. 5 В, 0..5 В, 0..10 В (переключается)	C200H-PID03
Модули позиционирования	1 ось	
	Импульсный выход; 1..100 000 Гц	C200H-NC111
	Импульсный выход; прямое подключение к приводу серводвигателя; 1..250 000 Гц	C200H-NC112
	Импульсный выход; прямое подключение к приводу серводвигателя; 1..500 000 Гц	C200HW-NC113
	2 оси	
	Импульсный выход; 1..250 000 Гц	C200H-NC113
	Импульсный выход; прямое подключение к приводу серводвигателя; 1..500 000 Гц	C200HW-NC213
	4 оси	
Импульсный выход; прямое подключение к приводу серводвигателя; 1..500 000 Гц	C200HW-NC413	
Модуль управления движением	2 оси, аналоговый выход	C200H-MC221
Модули высокоскоростных счетчиков	1 ось	
	Импульсный вход; скорость счета: 50 кГц	C200H-CT001-V1
	Импульсный вход; скорость счета: 75 кГц	C200H-CT002
	2 оси	
	Импульсный вход; скорость счета: 75 кГц	C200H-CT021
Модуль ASCII	Встроены RAM 24 К байт и 24 К байт EEPROM	C200H-ASC02
Модули датчика ID	Локальное применение, электромагнитное сцепление	C200H-IDS01-V 1
	Удаленное применение, микроволновая передача	C200H-IDS21
	Головки чтения/записи	
	Электромагнитный тип	серия V600-H
	Микроволновой тип	серия V620-H
	Носители данных	
	SRAM для серии V600-H	V600-D??R??
	EEPROM для серии V600-H	V600-D??R??
Модуль записи голосовых сообщений	Макс. 60 сообщений; длина сообщений; 32, 48 или 64 с (переключается)	C200H-OV001
	Кабель	
	RS-232C	C200H—CN224

**Замечание** При установке Модуля высокой плотности в качестве Модуля специальных входов/выходов на ведомой паенли, удаленный ведущий должен быть C200H-RM001-PV1 или C200H-RM201.

**Коммуникационные Модули**

Наименование	Характеристика		№ модели
Модуль SYSMAC LINK (коаксиальный кабель)	Таблица DATA LINK: 918 слов		C200HW-SLK23
	Таблица DATA LINK: 2 996 слов		C200HW-SLK24
	Терминатор	Требуется один для каждого узла на концах системы	C1000H-TER01
	Скоба крепления	Поставляется с Модулем SYSMAC LINK	C200H-TL01
	F-адаптер	Для подключения сети	C1000H-CE001
	Крышка F-адаптера	Для подключения сети	C1000H-COV01
Модуль SYSMAC LINK (оптоволоконный кабель H-PCF)	Таблица DATA LINK: 918 слов		C200HW-SLK13
	Таблица DATA LINK: 2 996 слов		C200HW-SLK14
	Адаптер питания для 1 и 2 Модулей	Требуется при питании от резервного напряжения.	C1000H-TER01
	Кабель питания		
	Для 1 Модуля	Соединяет адаптер питания и Модуль SYSMAC LINK.	C200H-CN111
	Для 2 Модулей		C200H-CN211
Плата SYSMAC LINK (коаксиальный кабель)	Для подключения IBM PC/AT или совместимый в качестве узла системы SYSMAC LINK.		3G8F5-SLK21-E
Модуль SYSMAC NET	Модуль соединения с шиной нужно заказывать отдельно		C200HW-SNT32
	Адаптер питания		
	Для 1 Модуля	Соединяет адаптер питания и Модуль SYSMAC NET	C200H-APS01
	Для 2 Модулей		C200H-APS02
	Кабель питания		
	Для 1 Модуля (Включено с C200H-APS01)	Соединяет адаптер питания и Модуль SYSMAC NET	C200H-CN001
Для 2 Модулей (Включено с C200H-APS02)	C200H-CN002		
Модули подключения шин	Для 1 Модуля	Соединяет Модуль SYSMAC LINK или SYSMAC NET с панелью связи C200HW-COM01/COM04-E	C200HW-CE001
	Для 2 Модулей		C200HW-CE002
Модули HOST LINK	APF/PCF		C200H-LK101-PV1
	RS-422		C200H-LK202-V1
	RS-232C		C200H-LK201-V1
Модуль PC LINK	Один уровень: 32 Модуля Несколько уровней: 16 Модулей	RS-485	C200H-LK401
Модули удаленных ведущих входов/выходов	До 2 на ПК; Можно подключить до 5 ведомых на ПК.	APF/PCF	C200H-RM001-PV1
		Витая пара	C200H-RM201
Модули удаленных ведомых входов/выходов	См. Панели в начале приложения.		

**Замечание** Модуль соединения с шиной нужно заказывать отдельно

**Оптоволоконный кабель для SYSMAC NET и SYSMAC LINK**  
**Оптоволоконный кабель H-PPCF с разъемами**

Наименование	Внешний вид	№ модели
SYSMAC NET SYSMAC LINK		S3200-CN???-20-20
		S3200-CN???-20-25
		S3200-CN???-20-62
		S3200-CN???-25-25
		S3200-CN???-25-62
		S3200-CN???-62-62

**Номера модели**

В вышеперечисленных номерах модели кабелей указываются тип кабеля, длина и тип разъема.

S3200- CN ???- 20-25

- 1, 2, 3,...** 1. S3200- CN определяет оптический кабель H-PCF  
 2 (???) заменяются кодами, указывающими длины стандартных моделей, как показано ниже.

О более длинных кабелях консультируйтесь с представителями OMRON в Вашем регионе. При заказе более длинных кабелей опускайте ту часть номера, где задается длина, и задавайте ее в метрах отдельно, например S3200-CN-20-20, 30 м.

Код	Длина, м	Код	Длина, м
201	2	152	15
501	5	202	20
102	10	Пропущен	более 20 м

3. Последние символы номера модели (например, 20-25) задает разъемы, как показано в таблице.

Код	Разъем
20	S3200-COCF2011
25	S3200-COCF2511
62	S3200-COCH62M

**Разъемы для оптоволоконных кабелей**

Номер модели/внешний вид	Подключаемые Модули	
	SYSMAC NET	SYSMAC LINK
S3200-COCF2011	CV500-SNT31	CV500-SLK11 C1000H-SLK11
S3200-COCF2511	C200HS-SNT32	C200HW-SLK13/14
S3200-COCH62M	S3200-LSU03-01E S3200-NSUA1-00E S3200-NSUG4-00E S3200-NSB11-E C500-SNT31-V4 B700-AL001	-

**Полностью пластмассовые оптоволоконные кабеля для SYSMAC BUS**

Наименование	Внешний вид	№ модели
Кабель	Только кабель; заказывайте требуемую длину 5..100 с интервалами 5 м, или с интервалами 200 м и 500 м.	3G5A2-PF002
Оптические разъемы А	Два оптических разъема (коричневые) для APF (10 м макс.)	3G5A2-CO001
Оптические разъемы В	Два оптических разъема (черные) для APF (8..10 м)	3G5A2-CO002
Комплект	Кабель 1 м с оптическим разъемом А на каждом конце	3G5A2-PF101

Наименование	Внешний вид	№ модели
Набор для обработки оптоволоконных кабелей	Принадлежности: 125-мм кусачки (MURUMOTO TEKKO'S 550M) для APF	3G5A2-CO002

**Оптоволоконные кабели в пластмассовой оболочке для SYSMAC BUS**

Наименование	Внешний вид		№ модели
Кабель (в помещении)	0.1 м с разъемами	Окружающая температура: -10..70 °С	3G5A2-OF011
	1 м с разъемами		3G5A2-OF 101
	2 м с разъемами		3G5A2-OF 201
	3 м с разъемами		3G5A2-OF 301
	5 м с разъемами		3G5A2-OF 501
	10 м с разъемами		3G5A2-OF 111
	20 м с разъемами		3G5A2-OF 211
	30 м с разъемами		3G5A2-OF 311
	40 м с разъемами		3G5A2-OF 411
	50 м с разъемами		3G5A2-OF 511
	Только кабель; заказывайте длину 1..500 м с интервалом 1 м		3G5A2-OF 002
	Только кабель; заказывайте длину 501..800 м с интервалом 1 м		Окружающая температура 0..55 °С (защищайте от прямого солнечного света)

**Оптоволоконные кабели H-PCF (для SYSMAC NET, SYSMAC LINK и SYSMAC BUS)**

Наименование	Характеристики		№ модели
Оптоволоконные кабели SYSMAC NET, SYSMAC LINK	10 м, черный	Композитный кабель, включая двухжильный кабель и двухжильный кабель питания	S3200-HCLB101
	50 м, черный		S3200-HCLB501
	100 м, черный		S3200-HCLB102
	500 м, черный		S3200-HCLB502
	1 000 м, черный		S3200-HCLB103
	10 м, оранжевый		S3200-HCLO101
	50 м, оранжевый		S3200-HCLO501
	100 м, оранжевый		S3200-HCLO102
	500 м, оранжевый		S3200-HCLO502
	1 000 м, оранжевый		S3200-HCLO103
Оптоволоконные кабели SYSMAC NET, SYSMAC LINK, SYSMAC BUS, SYSMAC WAY	10 м, черный	Двухжильный кабель	S3200-HCCB101
	50 м, черный		S3200-HCCB501
	100 м, черный		S3200-HCCB102
	500 м, черный		S3200-HCCB502
	1 000 м, черный		S3200-HCCB103
	10 м, оранжевый		S3200-HCCO101
	50 м, оранжевый		S3200-HCCO501
	100 м, оранжевый		S3200-HCCO102
	500 м, оранжевый		S3200-HCCO502
	1 000 м, оранжевый		S3200-HCCO103
	10 м, черный	Двухжильный кабель	S3200-HBCB101
	50 м, черный		S3200-HBCB501
	100 м, черный		S3200-HBCB102
	500 м, черный		S3200-HBCB502

Наименование	Характеристики	№ модели
	1 000 м, черный	S3200-HBCB103

**Разъемы оптоволоконного кабеля**

Наименование	Характеристики	№ модели
SYSMAC NET: S3200-LSU03-01E B700-AL001 C500-SNT31-V4	Разъем "замок" для Модулей SYSMAC NET NSU, NSB и C500	S3200-COCH62M
SYSMAC BUS: C200H-RM001-PV1 C200H-RT001/RT002-P C500-RM001-(P)V1 C500-RT001/RT002-(P) V1 3G2A9-??(-P)	Разъем "полузамок" для удаленных ведущих Модулей/выходов, удаленных ведомых, Модуля HOST LINK и адаптера связи	S3200-COCH82
SYSMAC NET /SYSMAC LINK : C200HS-SNT31 C200HW-SLK13/14	Разъем "полузамок"	S3200-COCF2511
SYSMAC NET /SYSMAC LINK : CV500-SNT31 CV500-SLK11 CV1000-SLK11	Разъем "замок"	S3200-COCF2011
Подключение ко всем узлам SYSMAC NET	COCF62M и COCF62F используются в паре	S3200-COCF62M S3200-COCF62F

**Замечание** Оптоволоконные кабеля должны быть подготовлены и подключены специалистами.

Если пользователь сам хочет готовить и подключать оптоволоконные кабеля, он должен пройти семинар SUMITOMO ELECTRIC INDUSTRIES и получить должную квалификацию.

Для подключения оптоволоконных кабелей требуются тестер оптического потока, Модуль головки, набор MASTER FIBER и устройство для сбора оптоволоконного кабеля.

Вы можете использовать Оптоволоконные кабеля в пластмассовой оболочке/пластиковые кабеля с разъемами, перечисленными ранее.

**Устройство для сбора оптоволоконного кабеля**

Наименование	Характеристики	№ модели
Устройство для сбора оптоволоконного кабеля	Служит для подключения разъемов H-PCF и обжимных разъемов для оптических систем связи, таких как SYSMAC BUS серии SYSMAC C- и CV-, SYSMAC LINK и SYSMAC NET.	S3200-CAK1062

**Замечание** 1. Оптоволоконные кабеля должны быть подготовлены и подключены специалистами.

2. Для подключения оптоволоконных кабелей требуются тестер оптического потока, Модуль головки, набор MASTER FIBER и устройство для сбора оптоволоконного кабеля.



**Тестер (проверочное устройство) оптического излучения**

Наименование	Характеристики	№ модели	
Тестер оптического излучения (см. прим.) (Поставляются с разъемом-адаптером, Модулем источником света, маленьким штекером с 1 головкой, жестким корпусом и адаптером пер. тока.)	SYSMAC NET: CV500-SNT31 C200HS-SNT32	C3200-CAT200 2 (поставляется с тестером)	S3200-CAT2002
	SYSMAC LINK : CV500-SLK11 C200HW-SLK13/14 CV1000-SLK11	C3200-CAT270 2 (поставляется с тестером)	S3200-CAT2702
	SYSMAC BUS: C200H-RM001-PV1 C200H-RT001/RT002-P C500-RM001-(P)V1 C500-RT001/RT002-(P)V1	C3200-CAT282 2 (поставляется с тестером)	S3200-CAT2822
	SYSMAC NET: S3200-LSU03-01E C500-SNT31-V4	C3200-CAT320 2 (поставляется с тестером)	S3200-CAT3202

**Замечание** Модуль источника света и переходник-разъем одинаков для Модуля головок и для тестера оптического излучения.

**Модуль головки**

Наименование	Характеристики	№ модели	
Модули головки (Разъем состоит из Модуля источника света и переходника-разъема) (см. прим.)	SYSMAC NET: CV500-SNT31 C200HS-SNT32		S3200-CAT2002
	SYSMAC LINK : CV500-SLK11 C200HW-SLK13/14 CV1000-SLK11		S3200-CAT2702
	SYSMAC BUS: C200H-RM001-PV1 C200H-RT001/RT002-P C500-RM001-(P)V1 C500-RT001/RT002-(P)V1		S3200-CAT2822
	SYSMAC NET: S3200-LSU03-01E C500-SNT31-V4		S3200-CAT3202

**Замечание** Пользуйтесь подходящей моделью Модуля головки для оптического модуля. Если используется два типа оптических модулей (тип Модуля и тип платы), заказывайте тестер оптического излучения плюс подходящую модель Модуля головки.

**MASTER - комплект оптоволоконна**

Наименование	Характеристики	№ модели
MASTER - комплект оптоволоконна (1 м)	S3200-CAT3202 (SYSMAC NET, NSB, NSU, BRIDGE)	S3200-CAT3201
	S3200-CAT2002/CAT2702 (SYSMAC NET, SYSMAC LINK)	S3200-CAT2001H
	S3200-CAT2822 (SYSMAC BUS )	S3200-CAT2821

**Замечание** Комплект оптоволоконна служит в комбинации с тестером оптического излучения для проверки оптического уровня оптоволоконного кабеля, подключенного к разъему.  
Оптоволоконные кабели должны быть подготовлены и подключены специалистами.  
Для подключения оптоволоконных кабелей требуются тестер оптического потока, Модуль головки, набор MASTER FIBER и устройство для сбора оптоволоконного кабеля.

### Программирующие устройства

Наименование	Характеристики	№ модели	
Консоль программирования	Портативный, с подсветкой	C200H-PRO27-E	
	Кабеля программатора	2 м	C200H-CN222
		4 м	C200H-CN422
	В комплекте с кабелем 2 м	CQM1-PRO01-E	
Скобы крепления программатора	Служит для крепления программатора к панели	C200H-ATT01	
Пульт задания параметров	Служит для ввода данных и индикации процесса для C200H-TC___, C200H-TV___, C200H-CP114 и C200H-PID0___	C200H-DSC01	
	Кабеля пульта задания параметров	2 м	C200H-CN225
		4 м	C200H-CN425
Соединительный кабель	Служит для подключения IBM PC/AT или совместимого к C200HX/HG/HE.	3.3 м CQM1-CIF02	

### Дополнительные устройства (опции)

Наименование	Характеристики	№ модели
Крышка Модуля входов/выходов	Крышка для клеммника на 10 клемм	C200H-COV11
Крышка Модуля клеммников	Защита для клеммника на 10 клемм (пакет из 10 крышек). 8 точек.	C200H-COV02
	Защита для клеммника на 19 клемм (пакет из 10 крышек). 12 точек.	C200H-COV02
Крышка разъема	Защитная крышка для неиспользуемых разъемов	C500H-SP001
Заглушка	Служит для установки на незанятые платоместа	C500H-COV01
Реле	24 В пост. тока	G6B-1174P-FD-US
Изоляционные пластины задней панели	Для C200HW-BC031 (задняя панель ЦПУ на 3 платоместа)	C200H-ATT31
	Для C200HW-BC051 (задняя панель ЦПУ на 5 платомест)	C200H-ATT51
	Для C200HW-BC081 (задняя панель ЦПУ на 8 платомест)	C200H-ATT81
	Для C200HW-BC101 (задняя панель ЦПУ на 10 платомест)	C200H-ATTA1
	Для C200HW-BI031 (задняя панель вх/вых на 3 платоместа)	C200H-ATT32
	Для C200HW-BI051 (задняя панель вх/вых на 5 платомест)	C200H-ATT52
	Для C200HW-BI081 (задняя панель вх/вых на 8 платомест)	C200H-ATT82
	Для C200HW-BI101 (задняя панель вх/вых на 10 платомест)	C200H-ATTA2
Крепления входов/выходов	Для задней панели на 3 платоместа	C200H-ATT33
	Для задней панели на 5 платомест	C200H-ATT53
	Для задней панели на 8 платомест	C200H-ATT83
	Для задней панели на 10 платомест	C200H-ATTA3
Внешние разъемы	Паяный разъем; 40 точек и крышка	C500-CE401
	Непаяный разъем; 40 точек и крышка; (обжим)	C500-CE402

Наименование	Характеристики	№ модели
	Разъем со сваркой под давлением; 40 точек	C500-CE403
	Паяный разъем; 40 точек и крышка (горизонтальный)	C500-CE404
	Обжимной разъем; 40 точек и крышка (горизонтальный)	C500-CE405

### Монтажные рейки и приспособления

Наименование	Характеристики	№ модели
Монтажная скоба DIN	1 комплект (2 штуки)	C200H-PR027-E
Рейка DIN	Длина: 50 см; высота: 7.3 см	PFP-50N
	Длина: 1 м; высота: 7.3 см	PFP-100N
	Длина: 1 м; высота: 16 мм	PFP-100N2
Конечная пластина		PFP-M
Заглушка		PFP-S

**Замечание** Заказывайте рейки DIN, защелки и заглушки наборами, кратными 10.

### Программное обеспечение

Наименование	Характеристики	№ модели
Программирование контроллера		
SYSMAC-CPT	Дискета 3.5", 2HD для IBM PC/IT (OC - Windows 95) Прим: Версия 1.0 не поддерживает дополнительные функции C200HX/HG/HE.	C500-ZL3AT1-E
Syswin v3.	5 Дискет 3.5", 2HD для IBM PC/IT (OC - )	SYSWIN-V3.2 NL
Пакет поддержки протокола		
	Дискета 3.5", 2HD для IBM PC/IT (OC - MS-DOS)	C200HW-ZW3AT1-E
	4 дискеты 3.5", 2HD для IBM PC/IT (OC - Windows 95)	WS01-PSTF1-E
Программное обеспечение для программирования Модулей позиционирования		
	4 дискеты 3.5", 2HD для IBM PC/IT (OC - Windows 95)	WS01-NCTF1-E
Программное обеспечение для программирования Модуля управления движением		
	Дискета 3.5", 2HD для IBM PC/IT (OC - MS-DOS)	CV500-ZN3AT1-E
	дискеты 3.5", 2HD для IBM PC/IT (OC - Windows 95)	SYSMAC-MST-V1.0-NL

### Материалы для обучения

Наименование	Характеристики	№ модели
Система обучения SYSMAC	Включает учебник, кассету и панель входных переключателей.	C200H-ETL01-E
Система обучения FUZZY	Включает Инструкцию Системы обучения FUZZY, Основной Модуль, Модуль памяти C200H—MR831, программатор C200H-PRO27-E, кабель программатора C200H-CN222, программу обучения FUZZY C500-SU981-E, кабель RS-232C и ремень для переноски.	C200H-ETL13-E



## Приложение В

### Команды релейно-контактного программирования

Команды ПК вводятся либо нажатием соответствующей клавиши на программаторе, (например, LD, AND, OR, NOT), либо используя функциональные коды. Для ввода команды с помощью ее функционального кода, нажмите FUN, функциональный код и WRITE. Подробности о программировании и командах смотри в пункте, указанном в правом столбце.

#### Команды C200HX/C200HG/C200HE-E

Код	Мнемоника	Наименование	Функция	Пункт
-	AND	AND	Логическое И состояния указанного бита и текущего условия исполнения.	5.8.1
-	AND LD	AND LOAD	Логическое И результатов предыдущих блоков	5.8.2
-	AND NOT	AND NOT	Логическое И инверсного состояния указанного бита и текущего условия исполнения.	5.8.1
-	CNT	COUNTER	Декрементальный счетчик	5.14.4
-	LD	LOAD	Служит для начала командной линии с состояния указанного бита или для задания логического блока для использования с AND LD и OR LD.	5.8.1
-	LD NOT	LOAD NOT	Служит для начала командной линии с инверсного состояния указанного бита.	5.8.1
-	OR	OR	Логическое ИЛИ состояния указанного бита и текущего условия исполнения.	5.8.1
-	OR LD	OR LOAD	Логическое ИЛИ результатов предыдущих блоков	5.8.2
-	OR NOT	OR NOT	Логическое ИЛИ инверсного состояния указанного бита и текущего условия исполнения.	5.8.1
-	OUT	OUTPUT	Включает в 1 битовый операнд при условии исполнения = 1. Включает в 0 битовый операнд при условии исполнения = 0.	5.9.1
-	OUT NOT	OUTPUT NOT	Включает в 0 битовый операнд при условии исполнения = 1. Включает в 1 битовый операнд при условии исполнения = 0. (т. е. операция инвертирования)	5.9.1
-	RSET	RESET	Выключает в 0 битовый операнд при условии исполнения = 1. Не влияет на состояние операнда при условии исполнения = 0.	5.9.3
-	SET	SET	Включает в 1 битовый операнд при условии исполнения = 1. Не влияет на состояние операнда при условии исполнения = 0.	5.9.3
-	TIM	TIMER	Операция задержки включения в 1 (декрементирующий таймер)	5.14.1
00	NOP	NO OPERATION	Ничего не выполняется и программа переходит к следующему адресу.	5.13
01	END	END	Требуется в конце программы	5.12
02	IL	INTERLOCK	Если условие исполнения = 0, то все выходы = 0, все текущие значения таймеров сбрасываются между IL и следующим IL.	5.10

**Приложение В. Команды релейно-контактного программирования**

Код	Мнемоника	Наименование	Функция	Пункт
03	ILC	INTERLOCK CLEAR	Другие операции воспринимаются как NOP. Текущие значения счетчиков сохраняются.	5.10
04	JMP	JUMP	Если условие исполнения = 0, все команды между JMP и	5.11
05	JME	JUMP END	соответствующей JMP игнорируются.	5.11
06	(@)FAL	FAILURE ALARM AND RESET	Генерирует признак нефатальной ошибки и выдает указанный номер FAL в программатор.	5.26.1
07	FALS	SEVERE FAILURE ALARM	Генерирует признак фатальной ошибки и выдает указанный номер FALS в программатор.	5.26.1
08	STEP	STEP DEFINE	При использовании с битом управления, назначает начало секции STEP и сбрасывает предыдущую секцию STEP. При использовании без N определяет окончание исполнения секции STEP.	5.25.1
09	SNXT	STEP START	Служит совместно с битом управления для указания окончания секции STEP и пуска следующей секции STEP.	5.25.1
10	SFT	SHIFT REGISTER	Создает битовый регистр сдвига.	5.15.1
11	KEEP	KEEP	Определяет бит - триггер, управляемый входами установки и сброса.	5.9.4
12	CNTR	REVERSIBLE COUNTER	Увеличивает или уменьшает текущее значение на 1, когда соответственно инкрементирующий и декрементирующий входы переключаются из 0 в 1.	5.14.5
13	DIFU	DIFFERENTIATE UP	Устанавливает в 1 указанный бит на один цикл при переднем фронте входного сигнала.	5.9.2
14	DIFD	DIFFERENTIATE DOWN	Устанавливает в 1 указанный бит на один цикл при заднем фронте входного сигнала.	5.9.2
15	TIMH	HIGH-SPEED TIMER	Высокоскоростной таймер с задержкой установки в 1 (декрементальный)	5.14.2
16	(@)WSFT	WORD SHIFT	Сдвиг данных между первым и последним словами блока, записывает нуль в первое слово.	5.15.9
17..19 Дополнительные команды				
20	CMP	COMPARE	Сравнивает содержание двух слов и выдает результат во флаги GR, EQ и LE.	5.17.2
21	(@)MOV	MOVE	Копирует данные (слово или константу) из источника в приемник.	5.16.1
22	(@)MVN	MOVE NOT	Инvertирует данные источника (слово или константу) и копирует в приемник.	5.16.2
23	(@)BIN	BCD-TO-BINARY	Преобразует 4-разрядное двоично-десятичное число источника в 16-битовое двоичное число и выдает преобразованное число в приемник.	5.18.1
24	(@)BCD	BINARY TO BCD	Преобразует двоичное число источника в двоично-десятичное и выдает преобразованное число в приемник.	5.18.3
25	(@)ASL	ARITHMETIC SHIFT LEFT	Сдвигает каждый бит одного слова на 1 бит влево, включая CY.	5.15.3
26	(@)ASR	ARITHMETIC SHIFT RIGHT	Сдвигает каждый бит одного слова на 1 бит вправо, включая CY.	5.15.4
27	(@)ROL	ROTATE LEFT	Осуществляет циклический сдвиг каждого бита одного слова на 1 бит влево, включая CY.	5.15.5
28	(@)ROR	ROTATE RIGHT	Осуществляет циклический сдвиг каждого бита одного слова на 1 бит влево, включая CY.	5.15.6
29	(@)COM	COMPLEMENT	Инvertирует состояние битов одного слова данных	5.23.1

**Приложение В. Команды релейно-контактного программирования**

Код	Мнемоника	Наименование	Функция	Пункт
30	(@)ADD	BCD ADD	Складывает два 4-разрядных двоично-десятичных числа и содержимое CY и выдает результат в указанное слово результата.	5.20.5
31	(@)SUB	BCD SUBTRACT	Вычитает одно 4-разрядное двоично-десятичное число и содержимое CY из другого 4-разрядного двоично-десятичного числа и выдает результат в указанное слово результата.	5.20.7
32	(@)MUL	BCD MULTIPLY	Перемножает два 4-разрядных двоично-десятичных числа и выдает результат в указанные слова результата.	5.20.9
33	(@)DIV	BCD DIVIDE	Делит одно 4-разрядное двоично-десятичное число на другое 4-разрядное двоично-десятичное число и выдает результат в указанные слова результата.	5.20.11
34	(@)ANDW	LOGICAL AND	Производит операцию ЛОГИЧЕСКОЕ И над двумя 16-битовыми входными словами и устанавливает соответствующие биты в слове результата, если оба входных бита = 1.	5.23.2
35	(@)ORW	LOGICAL OR	Производит операцию ЛОГИЧЕСКОЕ ИЛИ над двумя 16-битовыми входными словами и устанавливает соответствующие биты в слове результата, если один или оба входных бита = 1.	5.23.3
36	(@)XORW	EXCLUSIVE OR	Производит операцию ЛОГИЧЕСКОЕ ИЛИ над двумя 16-битовыми входными словами и устанавливает соответствующие биты в слове результата, если входные биты имеют разное значение.	5.23.4
37	(@)XNRW	EXCLUSIVE NOR	Производит операцию ЛОГИЧЕСКОЕ ИЛИ над двумя 16-битовыми входными словами и устанавливает соответствующие биты в слове результата, если входные биты имеют одинаковое значение.	5.23.5
38	(@)INC	INCREMENT	Инкрементирует (увеличивает на 1) 4-разрядное двоично-десятичное число.	5.20.1
39	(@)DEC	BCD DECREMENT	Декрементирует (уменьшает на 1) 4-разрядное двоично-десятичное число.	5.20.2
40	(@)STC	SET CARRY	Установить флаг переноса (CY) в 1	5.20.3
41	(@)CLC	CLEAR CARRY	Очистить флаг переноса (CY) в 0	5.20.4
45	TRSM	TRACE MEMORY SAMPLE	Пуск трассировки данных. Нельзя использовать с CQM1-CPU11/21-E или CPM1.	5.26.3
46	(@)MSG	MESSAGE	Отображает сообщение из 16 символов на дисплее программатора.	5.26.4
47..48 Дополнительные команды				
50	(@)ADB	Binary ADD	Складывает два 4-разрядных 16-ричных числа и содержимое CY и выдает результат в указанное слово результата.	5.21.1
51	(@)SBB	BINARY SUBTRACT	Вычитает одно 4-разрядное 16-ричное число и содержимое CY из другого 4-разрядного 16-ричного числа и выдает результат в указанное слово результата.	5.21.2
52	(@)MLB	BINARY MULTIPLY	Перемножает два 4-разрядных 16-ричных числа и выдает результат в указанные слова результата.	5.21.3
53	(@)DVB	BINARY DIVIDE	Делит одно 4-разрядное 16-ричное число на другое 4-разрядное 16-ричное число и выдает результат в указанные слова результата.	5.21.4

**Приложение В. Команды релейно-контактного программирования**

Код	Мнемоника	Наименование	Функция	Пункт
54	(@)ADDL	DOUBLE BCD ADD	Складывает два 8-разрядных (длиной 2 слова) двоично-десятичных числа и содержимое CY и выдает результат в указанное слово результата.	5.20.6
55	(@)SUBL	DOUBLE BCD SUBTRACT	Вычитает одно 8-разрядное (длиной 2 слова) двоично-десятичное число и содержимое CY из другого 8-разрядного (длиной 2 слова) двоично-десятичного числа и выдает результат в указанное слово результата.	5.20.8
56	(@)MULL	DOUBLE BCD MULTIPLY	Перемножает два 8-разрядных двоично-десятичных числа и выдает результат в указанные слова результата.	5.20.10
57	(@)DIVL	DOUBLE BCD DIVIDE	Делит одно 8-разрядное двоично-десятичное число на другое 8-разрядное двоично-десятичное число и выдает результат в указанные слова результата.	5.20.12
58	(@)BINL	DOUBLE BCD-TO-DOUBLE BINARY	Преобразует 8-разрядное двоично-десятичное число двух слов источника в двоичное число и выдает преобразованное число в два последовательных слова результата. (Только CQM1).	5.18.2
59	(@)BCDL	DOUBLE BINARY-TO-DOUBLE BCD	Преобразует двоичное число из двух слов источника в двоично-десятичное и выдает преобразованное число в два последовательных слова результата. (Только CQM1).	5.18.4
60..69 Дополнительные команды				
70	(@)XFER	BLOCK TRANSFER	Переносит содержание нескольких последовательных слов источника в несколько последовательных слов приемника.	5.16.4
71	(@)BSET	BLOCK SET	Переносит содержание константы или одного слова в несколько последовательных слов приемника.	5.16.3
72	(@)ROOT	SQUARE ROOT	Вычисляет квадратный корень за 8-разрядного двоично-десятичного числа и выдает 4-разрядный результат, округленный до целого, в указанное слово результата (только CQM1).	5.20.14
73	(@)XCHG	DATA EXCHANGE	Обменивает содержимое двух указанных слов	5.16.5
74	(@)SLD	ONE DIGIT SHIFT LEFT	Сдвигает влево данные между первым и последним словом на одну цифру (4 бита).	5.15.7
75	(@)SRD	ONE DIGIT SHIFT RIGHT	Сдвигает вправо данные между первым и последним словом на одну цифру (4 бита).	5.15.8
76	(@)MLPX	4-TO-16 DECODER	Преобразует до четырех 16-ричных цифр приемника в десятичные значения от 0 до 15 и устанавливает в слове (словах) результата соответствующие биты.	5.18.7
77	(@)DMPX	16-TO-4 ENCODER	Определяет позицию старшего бита в состоянии 1 в слове (словах) источника и включает в 1 соответствующие биты в слове результата.	5.18.8
78	(@)SDEC	7-SEGMENT DECODER	Преобразует 16-ричное значение из слова источника в данные для 7-сегментного индикатора.	5.18.9
79	(@)FDIV	FLOATING POINT DIVIDE	Делит одно число с плавающей точкой (DD+1, D) на другое (DR+1, DR) и выдает результат в R+1, R.	5.20.13
80	(@)DIST	SINGLE WORD DISTRIBUTE	Посылает слово источника в слово приемника, адрес которого задается в слове базы приемника плюс смещение.	5.16.6



## Приложение В. Команды релейно-контактного программирования

Код	Мнемоника	Наименование	Функция	Пункт
81	(@)COLL	DATA COLLECT	Извлекает данные из слова источника и записывает в слово приемника.	5.16.7
82	(@)MOVb	MOVE BIT	Пересылает указанный бит слова источника или константы в указанный бит слова приемника.	5.16.8
83	(@)MOVD	MOVE DIGIT	Пересылает 16-ричное значение указанной цифры (цифр) источника в указанные цифры (цифры) слова приемника (всего до 4 цифр).	5.16.9
84	(@)SFTR	REVERSIBLE SHIFT REGISTER	Сдвигает данные в указанном слове или серии слов влево либо вправо.	5.15.2
85	(@)TCMP	TABLE COMPARE	Сравнивает 4-разрядное 16-ричное значение с значениями таблицы, в которой 16 слов.	5.17.5
86	(@)ASC	ASCII CONVERT	Преобразование 16-ричное значение источника в 8-битный код ASCII, начиная от старшей либо младшей половины слова.	5.18.10
87.. 89 Дополнительные команды				
90	(@)SEND	NETWORK SEND	Служит для связи с другими ПК, связанными по сети SYSMAC NET или SYSMAC LINK.	5.27.1
91	(@)SBS	SUBROUTINE ENTRY	Вызов и исполнение подпрограммы N	5.24.3
92	SBN	SUBROUTINE DEFINE	Отметка начала подпрограммы N	5.24.4
93	RET	SUBROUTINE RETURN	Отметка конца подпрограммы N	5.24.4
94	(@)WDT	WATCH-DOG TIMER REFRESH	Увеличивает текущее значение контрольного таймера на 0..6300 мс.	5.26.7
97	(@)IORF	I/O REFRESH	Обновляет все слова входа/выхода между начальным и конечным словом	5.26.8
98	(@)RECV	NETWORK RECEIVE	Служит для связи с другими ПК, связанными по сети SYSMAC NET или SYSMAC LINK.	5.27.2
99	(@)MCRO	MACRO	Вызывает и исполняет подпрограмму с заменой слов входов/выходов	5.24.5

### Дополнительные команды

В таблице приведены дополнительные команды. Перед командами, у которых есть функциональные коды по умолчанию, приводятся эти коды по умолчанию.

Код	Мнемоника	Наименование	Функция	Пункт
17	(@)ASFT	ASYNCHRONOUS SHIFT REGISTER	Создает регистр сдвига, который обменивает содержимое соседних слов, когда одно из слов = 0, а другое нет.	5.15.10
18	(@)SCAN	CYCLE TIME	Задаёт минимальное время цикла (0..999.0 с)	5.26.2
19	(@)MCMP	MULTI-WORD COMPARE	Сравнивает блок из 16 последовательных слов с другим блоком из 16 последовательных слов.	5.17.1
47	(@)LMSG	32-CHARACTER MESSAGE	Выдает сообщение 32 символа на программатор	5.26.5
48	(@)TERM	TERMINAL MODE	Переключает программатор в режим TERMINAL для операции "отображение состояния клавиатуры"	5.26.6
60	CMPL	DOUBLE COMPARE	Сравнивает два 8-разрядных 16-ричных числа	5.17.3
61	(@)MPRF	GROUP-2 HIGH-DENSITY I/O REFRESH	Обновляет слова входов/выходов, выделенные блоком входов/выходов высокой плотности группы 2.	5.26.9
62	(@)XFRB	TRANSFER BITS	Копирует состояние до 255 заданных бит источника в указанные биты приемника.	5.16.10
63	(@)LINE	COLUMN TO LINE	Копирует столбец битов из 16 последовательных слов в указанное слово.	5.18.13
64	(@)COLM	LINE TO COLUMN	Копирует 16 битов из указанного слова в столбец битов 16 последовательных слов.	5.18.14

**Приложение В. Команды релейно-контактного программирования**

Код	Мнемоника	Наименование	Функция	Пункт
65	(@)SEC	HOURS TO SECONDS	Преобразует данные в форме часы и минуты в форму секунд.	5.18.5
66	(@)HMS	SECOND TO HOURS	Преобразует данные в секундах в данные в виде часов и минут.	5.18.6
67	(@)BCNT	BIT COUNTER	Подсчитывает общее число битов в состоянии 1 в указанном блоке слов.	5.26.10
68	(@)BCMP	BLOCK COMPARE	Определяет, находится ли значение слова в одной из 16 зон (задаются верхней и нижней границами).	5.17.4
69	(@)APR	ARITHMETIC PROCESS	Вычисляет синус, косинус, или линейную аппроксимацию.	5.22.5
87	TTIM	TOTALIZING TIMER	Создает суммирующий таймер	5.14.3
88	ZCP	AREA RANGE COMPARE	Сравнивает слово с зоной, заданной нижней и верхней границами и выдает результат в флаги GR, EQ и LE.	5.17.6
89	(@)INT	INTERRUPT CONTROL	Осуществляет управление прерываниями, маскирование и размаскирование входных прерываний.	5.24.6
-	7SEG	7-SEGMENT DISPLAY OUTPUT	Преобразует 4- или 8-разрядные данные в формат 7-сегментного индикатора и выводит преобразованные данные.	5.29.5
-	(@)ADBL	DOUBLE BINARY ADD	Складывает два 8-разрядных двоичных чисел двойной длины (со знаком или без знака) и выдает результат в R и R+1.	5.21.5
-	AVG	AVERAGE VALUE	Складывает указанное количество 16-ричных слов и вычисляет среднее значение. Округляет до 4 цифр после запятой.	5.22.3
-	(@)BXF2	EM BANK TRANSFER	Переносит содержимое нескольких последовательных слов источника в несколько последовательных слов приемника. Слова в текущем банке EM можно задать для источнике или приемника.	5.16.12
-	CPS	SIGNED BINARY COMPARE	Сравнивает два 16-битовых (4 цифры) двоичных чисел со знаком и выдает результат во флаги GR, EQ и LE.	5.17.8
-	CPSL	DOUBLE SIGNED BINARY COMPARE	Сравнивает два 32-битовых (8 цифр) двоичных чисел двойной длины со знаком и выдает результат во флаги GR, EQ и LE.	5.17.9
-	(@)DBS	SIGNED BINARY DIVIDE	Делит одно 16-битовое двоичное число со знаком на другое и выдает 32-битовый двоичный результат со знаком в R+1 и R.	5.21.9
-	(@)DBSL	DOUBLE SIGNED BINARY DIVIDE	Делит одно 32-битовое двоичное число со знаком на другое и выдает 64-битовый двоичный результат со знаком в R+3..R	5.21.10
-	DSW	DIGITAL SWITCH INPUT	Вводит 4- или 8-разрядные двоично-десятичные данные с кодового колеса.	5.29.1
-	(@)EMBC	SELECT EM BANK	Изменяет текущий банк EM на номер заданного банка.	5.26.16
-	(@)FCS	FCS CALCULATE	Проверяет ошибки в данных, переданных в запросе при связи с верхним уровнем.	5.26.11
-	FPD	FAILURE POINT DETECT	Ищет сбои в блоке команд.	5.25.12
-	(@)HEX	ASCII-TO-HEXADECIMAL	Преобразует данные ASCII в 16-ричные.	5.18.11
-	HKY	HEXADECIMAL KEY INPUT	Вводит до 8 цифр 16-ричных данных с клавиатуры 16 клавиш.	5.29.3

**Приложение В. Команды релейно-контактного программирования**

Код	Мнемоника	Наименование	Функция	Пункт
-	(@)IEMS	INDIRECT EM ADDRESSING	Изменяет адерата косвенной адресации DM (*DM) на DM или указанный банк EM. Данная команда может использоваться для изменения текущего банка EM.	5.26.15
-	(@)IORD	SPECIAL I/O UNIT READ	Передает данные из памяти указанных блоков специальных входов/выходов в слова ПК.	5.30.1
-	(@)IOWR	SPECIAL I/O UNIT WRITE	Передает данные из слов ПК в памяти указанных блоков специальных входов/выходов.	5.30.2
-	(@)MAX	FIND MAXIMUM	Находит максимальное значение в указанной области данных и выдает это значение в другое слово.	5.22.1
-	(@)MBS	SIGNED BINARY MULTIPLY	Умножает два 16-битовых двоичных числа со знаком и выдает 8-разрядный двоичный результат со знаком в R+1 и R.	5.21.7
-	(@)MBSL	DOUBLE SIGNED BINARY MULTIPLY	Умножает два 32-битовых двоичных числа двойной длины со знаком и выдает 16-разрядный двоичный результат со знаком в R+3..R.	5.21.8
-	(@)MIN	FIND MINIMUM	Находит минимальное значение в указанной области данных и выдает это значение в другое слово.	5.22.2
-	MTR	MATRIX INPUT	Вводит данные с матрицы 8 точек входа x 8 точек выхода.	5.29.4
-	(@)NEG	2'S COMPLEMENT	Преобразует 4-разрядное 16-ричное содержимое источника в дополнение до 2 и выдает в результат в R.	5.18.15
-	(@)NEGL	DOUBLE 2'S COMPLEMENT	Преобразует 8-разрядное 16-ричное содержимое слов источника в дополнение до 2 и выдает в результат в R и R+1.	5.18.16
-	(@)PID	PID CONTROL	Осуществляет ПИД-регулирование, основанное на предварительно заданных параметрах.	5.22.6
-	(@)PMCR	PROTOCOL MACRO	Вызывает и выполняет заданный протокол связи, который зарегистрирован в панели связи.	5.28.4
-	(@)RXD	RECEIVE	Получает данные через порт связи	5.28.1
-	(@)SBBL	DOUBLE BINARY SUBTRACT	Вычитает одно 8-разрядное двоичное число (со знаком или без) из другого и выдает результат в R+1 и R.	5.21.6
-	(@)SCL	SCALING	Осуществляет масштабирование вычисляемой величины.	5.18.12
-	(@)SRCH	DATA SEARCH	Ищет в заданной зоне памяти указанные данные. Выдает адрес (адреса) слов, которые содержат данные.	5.26.13
-	(@)STUP	CHAMGE RS-232C SETUP	Изменение установочных параметров указанного порта.	5.28.3
-	(@)SUM	SUM	Подсчитывает сумму содержимого слов в указанной зоне памяти.	5.22.4
-	(@)TKY	TEN KEY INPUT	Вводит 8 двоично-десятичных цифр из клавиатуры на 10 клавиш.	5.29.2
-	(@)TXD	TRANSMIT	Посылает данные через порт связи	5.28.2
-	(@)XDMR	EXPANSION DM READ	Содержимое указанного числа слов области расширенных DM читаются и выдаются в слова приемника ПК.	5.26.14

**Приложение В. Команды релейно-контактного программирования**

Код	Мнемоника	Наименование	Функция	Пункт
-	(@)XFR2	EM BLOCK TRANSFER	Переносит содержимое нескольких последовательных слов источника в несколько последовательных слов приемника. Для источника или приемника можно задавать слова в любом действующем банке EM. Команду можно использовать для изменения текущего банка.	5.16.11
-	ZCPL	DOUBLE AREA RANGE COMPARE	Сравнивает 8-разрядное значение с зоной, заданной нижней и верхней границами и выдает результат в флаги GR, EQ и LE.	5.17.7

**Команды C200HX/C200HG/C200HE-E**

Код	Мнемоника	Наименование	Функция	Пункт
-	AND	AND	Логическое И состояния указанного бита и текущего условия исполнения.	5.8.1
-	AND LD	AND LOAD	Логическое И результатов предыдущих блоков	5.8.2
-	AND NOT	AND NOT	Логическое И инверсного состояния указанного бита и текущего условия исполнения.	5.8.1
-	CNT	COUNTER	Декрементальный счетчик	5.14.4
-	LD	LOAD	Служит для начала командной линии с состояния указанного бита или для задания логического блока для использования с AND LD и OR LD.	5.8.1
-	LD NOT	LOAD NOT	Служит для начала командной линии с инверсного состояния указанного бита.	5.8.1
-	OR	OR	Логическое ИЛИ состояния указанного бита и текущего условия исполнения.	5.8.1
-	OR LD	OR LOAD	Логическое ИЛИ результатов предыдущих блоков	5.8.2
-	OR NOT	OR NOT	Логическое ИЛИ инверсного состояния указанного бита и текущего условия исполнения.	5.8.1
-	OUT	OUTPUT	Включает в 1 битовый операнд при условии исполнения = 1. Включает в 0 битовый операнд при условии исполнения = 0.	5.9.1
-	OUT NOT	OUTPUT NOT	Включает в 0 битовый операнд при условии исполнения = 1. Включает в 1 битовый операнд при условии исполнения = 0. (т. е. операция инвертирования)	5.9.1
-	RSET	RESET	Выключает в 0 битовый операнд при условии исполнения = 1. Не влияет на состояние операнда при условии исполнения = 0.	5.9.3
-	SET	SET	Включает в 1 битовый операнд при условии исполнения = 1. Не влияет на состояние операнда при условии исполнения = 0.	5.9.3
-	TIM	TIMER	Операция задержки включения в 1 (декрементирующий таймер)	5.14.1
000	NOP	NO OPERATION	Ничего не выполняется и программа переходит к следующему адресу.	5.13
001	END	END	Требуется в конце программы	5.12
002	IL	INTERLOCK	Если условие исполнения = 0, то все выходы = 0, все текущие значения таймеров сбрасываются между IL и следующим IL.	5.10
003	ILC	INTERLOCK CLEAR	Другие операции воспринимаются как NOP. Текущие значения счетчиков сохраняются.	5.10

**Приложение В. Команды релейно-контактного программирования**

Код	Мнемоника	Наименование	Функция	Пункт
004	JMP	JUMP	Если условие исполнения = 0, все команды между JMP и	5.11
005	JME	JUMP END	соответствующей JMP игнорируются.	5.11
006	(@)FAL	FAILURE ALARM AND RESET	Генерирует признак нефатальной ошибки и выдает указанный номер FAL в программатор.	5.26.1
007	FALS	SEVERE FAILURE ALARM	Генерирует признак фатальной ошибки и выдает указанный номер FALS в программатор.	5.26.1
008	STEP	STEP DEFINE	При использовании с битом управления, назначает начало секции STEP и сбрасывает предыдущую секцию STEP. При использовании без N определяет окончание исполнения секции STEP.	5.25.1
009	SNXT	STEP START	Служит совместно с битом управления для указания окончания секции STEP и пуска следующей секции STEP.	5.25.1
010	SFT	SHIFT REGISTER	Создает битовый регистр сдвига.	5.15.1
011	KEEP	KEEP	Определяет бит - триггер, управляемый входами установки и сброса.	5.9.4
012	CNTR	REVERSIBLE COUNTER	Увеличивает или уменьшает текущее значение на 1, когда соответственно инкрементирующий и декрементирующий входы переключаются из 0 в 1.	5.14.5
013	DIFU	DIFFERENTIATE UP	Устанавливает в 1 указанный бит на один цикл при переднем фронте входного сигнала.	5.9.2
014	DIFD	DIFFERENTIATE DOWN	Устанавливает в 1 указанный бит на один цикл при заднем фронте входного сигнала.	5.9.2
015	TIMH	HIGH-SPEED TIMER	Высокоскоростной таймер с задержкой установки в 1 (декрементальный)	5.14.2
016	(@)WSFT	WORD SHIFT	Сдвиг данных между первым и последним словами блока, записывает нуль в первое слово.	5.15.9
017	(@)ASFT	ASYNCHRONOUS SHIFT REGISTER	Создает регистр сдвига, который обменивает содержимое соседних слов, когда одно из слов = 0, а другое нет.	5.15.10
018	(@)SCAN	CYCLE TIME	Задаёт минимальное время цикла (0..999.0 с)	5.26.2
019	(@)MCMP	MULTI-WORD COMPARE	Сравнивает блок из 16 последовательных слов с другим блоком из 16 последовательных слов.	5.17.1
020	CMP	COMPARE	Сравнивает содержание двух слов и выдает результат во флаги GR, EQ и LE.	5.17.2
021	(@)MOV	MOVE	Копирует данные (слово или константу) из источника в приемник.	5.16.1
022	(@)MVN	MOVE NOT	Инвертирует данные источника (слово или константу) и копирует в приемник.	5.16.2
023	(@)BIN	BCD-TO-BINARY	Преобразует 4-разрядное двоично-десятичное число источника в 16-битовое двоичное число и выдает преобразованное число в приемник.	5.18.1
024	(@)BCD	BINARY TO BCD	Преобразует двоичное число источника в двоично-десятичное и выдает преобразованное число в приемник.	5.18.3
025	(@)ASL	ARITHMETIC SHIFT LEFT	Сдвигает каждый бит одного слова на 1 бит влево, включая CY.	5.15.3
026	(@)ASR	ARITHMETIC SHIFT RIGHT	Сдвигает каждый бит одного слова на 1 бит вправо, включая CY.	5.15.4
027	(@)ROL	ROTATE LEFT	Осуществляет циклический сдвиг каждого бита одного слова на 1 бит влево, включая CY.	5.15.5
028	(@)ROR	ROTATE RIGHT	Осуществляет циклический сдвиг каждого бита одного слова на 1 бит вправо, включая CY.	5.15.6

**Приложение В. Команды релейно-контактного программирования**

Код	Мнемоника	Наименование	Функция	Пункт
029	(@)COM	COMPLEMENT	Инвертирует состояние битов одного слова данных	5.23.1
030	(@)ADD	BCD ADD	Складывает два 4-разрядных двоично-десятичных числа и содержимое CY и выдает результат в указанное слово результата.	5.20.5
031	(@)SUB	BCD SUBTRACT	Вычитает одно 4-разрядное двоично-десятичное число и содержимое CY из другого 4-разрядного двоично-десятичного числа и выдает результат в указанное слово результата.	5.20.7
032	(@)MUL	BCD MULTIPLY	Перемножает два 4-разрядных двоично-десятичных числа и выдает результат в указанные слова результата.	5.20.9
033	(@)DIV	BCD DIVIDE	Делит одно 4-разрядное двоично-десятичное число на другое 4-разрядное двоично-десятичное число и выдает результат в указанные слова результата.	5.20.11
034	(@)ANDW	LOGICAL AND	Производит операцию ЛОГИЧЕСКОЕ И над двумя 16-битовыми входными словами и устанавливает соответствующие биты в слове результата, если оба входных бита = 1.	5.23.2
035	(@)ORW	LOGICAL OR	Производит операцию ЛОГИЧЕСКОЕ ИЛИ над двумя 16-битовыми входными словами и устанавливает соответствующие биты в слове результата, если один или оба входных бита = 1.	5.23.3
036	(@)XORW	EXCLUSIVE OR	Производит операцию ЛОГИЧЕСКОЕ ИЛИ над двумя 16-битовыми входными словами и устанавливает соответствующие биты в слове результата, если входные биты имеют разное значение.	5.23.4
037	(@)XNRW	EXCLUSIVE NOR	Производит операцию ЛОГИЧЕСКОЕ ИЛИ над двумя 16-битовыми входными словами и устанавливает соответствующие биты в слове результата, если входные биты имеют одинаковое значение.	5.23.5
038	(@)INC	INCREMENT	Инкрементирует (увеличивает на 1) 4-разрядное двоично-десятичное число.	5.20.1
039	(@)DEC	BCD DECREMENT	Декрементирует (уменьшает на 1) 4-разрядное двоично-десятичное число.	5.20.2
040	(@)STC	SET CARRY	Установить флаг переноса (CY) в 1	5.20.3
041	(@)CLC	CLEAR CARRY	Очистить флаг переноса (CY) в 0	5.20.4
045	TRSM	TRACE MEMORY SAMPLE	Пуск трассировки данных.	5.26.3
046	(@)MSG	MESSAGE	Отображает сообщение из 16 символов на дисплее программатора.	5.26.4
047	(@)LMSG	32-CHARACTER MESSAGE	Выдает сообщение 32 символа на программатор	5.26.5
048	(@)TERM	TERMINAL MODE	Переключает программатор в режим TERMINAL для операции "отображение состояния клавиатуры"	5.26.6
050	(@)ADB	Binary ADD	Складывает два 4-разрядных 16-ричных числа и содержимое CY и выдает результат в указанное слово результата.	5.21.1
051	(@)SBB	BINARY SUBTRACT	Вычитает одно 4-разрядное 16-ричное число и содержимое CY из другого 4-разрядного 16-ричного числа и выдает результат в указанное слово результата.	5.21.2

**Приложение В. Команды релейно-контактного программирования**

Код	Мнемоника	Наименование	Функция	Пункт
052	(@)MLB	BINARY MULTIPLY	Перемножает два 4-разрядных 16-ричных числа и выдает результат в указанные слова результата.	5.21.3
053	(@)DVB	BINARY DIVIDE	Делит одно 4-разрядное 16-ричное число на другое 4-разрядное 16-ричное число и выдает результат в указанные слова результата.	5.21.4
054	(@)ADDL	DOUBLE BCD ADD	Складывает два 8-разрядных (длиной 2 слова) двоично-десятичных числа и содержимое СУ и выдает результат в указанное слово результата.	5.20.6
055	(@)SUBL	DOUBLE BCD SUBTRACT	Вычитает одно 8-разрядное (длиной 2 слова) двоично-десятичное число и содержимое СУ из другого 8-разрядного (длиной 2 слова) двоично-десятичного числа и выдает результат в указанное слово результата.	5.20.8
056	(@)MULL	DOUBLE BCD MULTIPLY	Перемножает два 8-разрядных двоично-десятичных числа и выдает результат в указанные слова результата.	5.20.10
057	(@)DIVL	DOUBLE BCD DIVIDE	Делит одно 8-разрядное двоично-десятичное число на другое 8-разрядное двоично-десятичное число и выдает результат в указанные слова результата.	5.20.12
058	(@)BINL	DOUBLE BCD-TO-DOUBLE BINARY	Преобразует 8-разрядное двоично-десятичное число двух слов источника в двоичное число и выдает преобразованное число в два последовательных слова результата..	5.18.2
059	(@)BCDL	DOUBLE BINARY-TO-DOUBLE BCD	Преобразует двоичное число из двух слов источника в двоично-десятичное и выдает преобразованное число в два последовательных слова результата.	5.18.4
060	CMPL	DOUBLE COMPARE	Сравнивает два 8-разрядных 16-ричных числа	5.17.3
061	(@)MPRF	GROUP-2 HIGH-DENSITY I/O REFRESH	Обновляет слова входов/выходов, выделенные блокам входов/выходов высокой плотности группы 2.	5.26.9
062	(@)XFRB	TRANSFER BITS	Копирует состояние до 255 заданных бит источника в указанные биты приемника.	5.16.10
063	(@)LINE	COLUMN TO LINE	Копирует столбец битов из 16 последовательных слов в указанное слово.	5.18.13
064	(@)COLM	LINE TO COLUMN	Копирует 16 битов из указанного слова в столбец битов 16 последовательных слов.	5.18.14
065	(@)SEC	HOURS TO SECONDS	Преобразует данные в форме часы и минуты в форму секунд.	5.18.5
066	(@)HMS	SECOND TO HOURS	Преобразует данные в секундах в данные в виде часов и минут.	5.18.6
067	(@)BCNT	BIT COUNTER	Подсчитывает общее число битов в состоянии 1 в указанном блоке слов.	5.26.10
068	(@)BCMP	BLOCK COMPARE	Определяет, находится ли значение слова в одной из 16 зон (задаются верхней и нижней границами).	5.17.4
069	(@)APR	ARITHMETIC PROCESS	Вычисляет синус, косинус, или линейную аппроксимацию.	5.22.5
070	(@)XFER	BLOCK TRANSFER	Переносит содержание нескольких последовательных слов источника в несколько последовательных слов приемника.	5.16.4
071	(@)BSET	BLOCK SET	Переносит содержание константы или одного слова в несколько последовательных слов приемника.	5.16.3

**Приложение В. Команды релейно-контактного программирования**

Код	Мнемоника	Наименование	Функция	Пункт
072	(@)ROOT	SQUARE ROOT	Вычисляет квадратный корень за 8-разрядного двоично-десятичного числа и выдает 4-разрядный результат, округленный до целого, в указанное слово результата (только CQM1 ).	5.20.14
073	(@)XCHG	DATA EXCHANGE	Обменивает содержимое двух указанных слов	5.16.5
074	(@)SLD	ONE DIGIT SHIFT LEFT	Сдвигает влево данные между первым и последним словом на одну цифру (4 бита).	5.15.7
075	(@)SRD	ONE DIGIT SHIFT RIGHT	Сдвигает вправо данные между первым и последним словом на одну цифру (4 бита).	5.15.8
076	(@)MLPX	4-TO-16 DECODER	Преобразует до четырех 16-ричных цифр приемника в десятичные значения от 0 до 15 и устанавливает в слове (словах) результата соответствующие биты.	5.18.7
077	(@)DMPX	16-TO-4 ENCODER	Определяет позицию старшего бита в состоянии 1 в слове (словах) источника и включает в 1 соответствующие биты в слове результата.	5.18.8
078	(@)SDEC	7-SEGMENT DECODER	Преобразует 16-ричное значение из слова источника в данные для 7-сегментного индикатора.	5.18.9
079	(@)FDIV	FLOATING POINT DIVIDE	Делит одно число с плавающей точкой (DD+1, D) на другое (DR+1, DR) и выдает результат в R+1, R.	5.20.13
080	(@)DIST	SINGLE WORD DISTRIBUTE	Посылает слово источника в слово приемника, адрес которого задается в слове базы приемника плюс смещение.	5.16.6
081	(@)COLL	DATA COLLECT	Извлекает данные из слова источника и записывает в слово приемника.	5.16.7
082	(@)MOVB	MOVE BIT	Пересылает указанный бит слова источника или константы в указанный бит слова приемника.	5.16.8
083	(@)MOVD	MOVE DIGIT	Пересылает 16-ричное значение указанной цифры (цифр) источника в указанные цифры (цифру) слова приемника (всего до 4 цифр).	5.16.9
084	(@)SFTR	REVERSIBLE SHIFT REGISTER	Сдвигает данные в указанном слове или серии слов влево либо вправо.	5.15.2
085	(@)TCMP	TABLE COMPARE	Сравнивает 4-разрядное 16-ричное значение с значениями таблицы, в которой 16 слов.	5.17.5
086	(@)ASC	ASCII CONVERT	Преобразование 16-ричное значение источника в 8-битный код ASCII, начиная от старшей либо младшей половины слова.	5.18.10
087	TTIM	TOTALIZING TIMER	Создает суммирующий таймер	5.14.3
088	(@)ZCP	AREA RANGE COMPARE	Сравнивает слово с зоной, заданной нижней и верхней границами и выдает результат в флаги GR, EQ и LE.	5.17.6
089	(@)INT	INTERRUPT CONTROL	Осуществляет управление прерываниями, маскирование и размаскирование входных прерываний.	5.24.6
090	(@)SEND	NETWORK SEND	Служит для связи с другими ПК, связанными по сети SYSMAC NET или SYSMAC LINK.	5.27.1
091	(@)SBS	SUBROUTINE ENTRY	Вызов и исполнение подпрограммы N	5.24.3
092	SBN	SUBROUTINE DEFINE	Отметка начала подпрограммы N	5.24.4
093	RET	SUBROUTINE RETURN	Отметка конца подпрограммы N	5.24.4
094	(@)WDT	WATCH-DOG TIMER REFRESH	Увеличивает текущее значение контрольного таймера на 0..6300 мс.	5.26.7
097	(@)IORF	I/O REFRESH	Обновляет все слова входа/выхода между начальным и конечным словом	5.26.8



**Приложение В. Команды релейно-контактного программирования**

Код	Мнемоника	Наименование	Функция	Пункт
098	(@)RECV	NETWORK RECEIVE	Служит для связи с другими ПК, связанными по сети SYSMAC NET или SYSMAC LINK.	5.27.2
099	(@)MCRO	MACRO	Вызывает и исполняет подпрограмму с заменой слов входов/выходов	5.24.5
114	(@)CPS	SIGNED BINARY COMPARE	Сравнивает два 16-битовых (4 цифры) двоичных чисел со знаком и выдает результат во флаги GR, EQ и LE.	5.17.8
115	(@)CPSL	DOUBLE SIGNED BINARY COMPARE	Сравнивает два 32-битовых (8 цифр) двоичных чисел двойной длины со знаком и выдает результат во флаги GR, EQ и LE.	5.17.9
116	(@)ZCPL	DOUBLE AREA RANGE COMPARE	Сравнивает 8-разрядное значение с зоной, заданной нижней и верхней границами и выдает результат в флаги GR, EQ и LE.	5.17.7
160	(@)NEG	2'S COMPLEMENT	Преобразует 4-разрядное 16-ричное содержимое источника в дополнение до 2 и выдает в результат в R.	5.18.15
161	(@)NEGL	DOUBLE 2'S COMPLEMENT	Преобразует 8-разрядное 16-ричное содержимое слов источника в дополнение до 2 и выдает в результат в R и R+1.	5.18.16
162	(@)HEX	ASCII-TO-HEXADECIMAL	Преобразует данные ASCII в 16-ричные.	5.18.11
180	(@)FCS	FCS CALCULATE	Проверяет ошибки в данных, переданных в запросе при связи с верхним уровнем.	5.26.11
181	(@)SRCH	DATA SEARCH	Ищет в заданной зоне памяти указанные данные. Выдает адрес (адреса) слов, которые содержат данные.	5.26.13
182	(@)MAX	FIND MAXIMUM	Находит максимальное значение в указанной области данных и выдает это значение в другое слово.	5.22.1
183	(@)MIN	FIND MINIMUM	Находит минимальное значение в указанной области данных и выдает это значение в другое слово.	5.22.2
184	(@)SUM	SUM	Подсчитывает сумму содержимого слов в указанной зоне памяти.	5.22.4
190	(@)PID	PID CONTROL	Осуществляет ПИД-регулирование, основанное на предварительно заданных параметрах.	5.22.6
194	(@)SCL	SCALING	Осуществляет масштабирование вычисляемой величины.	5.18.12
195	AVG	AVERAGE VALUE	Складывает указанное количество 16-ричных слов и вычисляет среднее значение. Округляет до 4 цифр после запятой.	5.22.3
210	DSW	DIGITAL SWITCH INPUT	Вводит 4- или 8-разрядные двоично-десятичные данные с кодового колеса.	5.29.1
211	(@)TKY	TEN KEY INPUT	Вводит 8 двоично-десятичных цифр из клавиатуры на 10 клавиш.	5.29.2
212	HKY	HEXADECIMAL KEY INPUT	Вводит до 8 цифр 16-ричных данных с клавиатуры 16 клавиш.	5.29.3
213	MTR	MATRIX INPUT	Вводит данные с матрицы 8 точек входа x 8 точек выхода.	5.29.4
214	7SEG	7-SEGMENT DISPLAY OUTPUT	Преобразует 4- или 8-разрядные данные в формат 7-сегментного индикатора и выводит преобразованные данные.	5.29.5
222	(@)IORD	SPECIAL I/O UNIT READ	Передает данные из памяти указанных блоков специальных входов/выходов в слова ПК.	5.30.1
223	(@)IOWR	SPECIAL I/O UNIT WRITE	Передает данные из слов ПК в память указанных блоков специальных входов/выходов.	5.30.2
235	(@)RXD	RECEIVE	Получает данные через порт связи	5.28.1

**Приложение В. Команды релейно-контактного программирования**

<b>Код</b>	<b>Мнемоника</b>	<b>Наименование</b>	<b>Функция</b>	<b>Пункт</b>
236	(@)TXD	TRANSMIT	Посылает данные через порт связи	5.28.2
237	(@)STUP	CHAMGE RS-232C SETUP	Изменение установочных параметров указанного порта.	5.28.3
260	(@)PMCR	PROTOCOL MACRO	Вызывает и выполняет заданный протокол связи, который зарегистрирован в панели связи.	5.28.4
261	(@)CMCR	PCMCIA CARD MACRO	Выполняет процедуры по записи, чтению, поиску и сравнению файлов в Модуле PC Card	5.26.17
269	FPD	FAILURE POINT DETECT	Ищет сбои в блоке команд.	5.26.12
280	(@)XDMR	EXPANSION DM READ	Содержимое указанного числа слов области расширенных DM читаются и выдаются в слова приемника ПК.	5.26.14
281	(@)EMBC	SELECT EM BANK	Изменяет текущий банк EM на номер заданного банка.	5.26.16
300	=	EQUAL		5.7.10
301	=L	DOUBLE EUQUAL		5.7.10
302	=S	SIGNED EQUAL		5.7.10
303	=SL	DOUBLE SIGNED EQUAL		5.7.10
305	<>	NOT EQUAL		5.7.10
306	<>L	DOUBLE NOT EQUAL		5.7.10
307	<>S	SIGNED NOT EQUAL		5.7.10
308	<>SL	DOUBLE SIGNED NOT EQUAL		5.7.10
310	<	LESS THAN		5.7.10
311	<L	DOUBLE SIGNED LESS THSN		5.7.10
312	<S	SIGNED LESS THAN		5.7.10
313	<SL	DOUBLE SIGNED LESS THAN		5.7.10
315	<=	LESS THAN OR EQUAL		5.7.10
316	<=L	DOUBLE LESS THAN OR EQUAL		5.7.10
317	<=S	SIGNED LESS THAN OR EQUAL		5.7.10
318	<=SL	DOUBLE SIGNED LESS THAN OR EQUAL		5.7.10
320	>	GREATER THAN		5.7.10
321	>L	DOUBLE GREATER THAN		5.7.10
322	>S	SIGNED GREATER THAN		5.7.10
323	>SL	DOUBLE SIGNED GREATER THAN		5.7.10
325	>=	GREATER THAN OR EQUAL		5.7.10
326	>=L	DOUBLE GREATER THAN OR EQUAL		5.7.10
327	>=S	SIGNED GREATER THAN OR EQUAL		5.7.10

**Приложение В. Команды релейно-контактного программирования**

<b>Код</b>	<b>Мнемоника</b>	<b>Наименование</b>	<b>Функция</b>	<b>Пункт</b>
328	>=SL	DOUBLE SIGNED GREATER THAN OR EQUAL		5.7.10
350	TST	TEST BIT		5.9.5
351	TSTN	TEST BIT NOT		5.9.5
400	(@)+	SIGNED BINARY ADD WITHOUT CARRY		5.19.1
401	(@)+L	DOUBLE SIGNED BINARY ADD WITHOUT CARRY		5.19.1
402	(@)+C	SIGNED BINARY ADD WITH CARRY		5.19.1
403	(@)+CL	DOUBLE SIGNED BINARY ADD WITH CARRY		5.19.1
404	(@)+B	BCD ADD WITHOUT CARRY		5.19.2
405	(@)+BL	DOUBLE BCD ADD WITHOUT CARRY		5.19.2
406	(@)+BC	BCD ADD WITH CARRY		5.19.2
407	(@)+BCL	DOUBLE BCD ADD WITH CARRY		5.19.2
410	(@)-	SIGNED BINARY SUBSTRACT WITHOUT CARRY		5.19.3
411	(@)-L	DOUBLE SIGNED BINARY SUBSTRACT WITHOUT CARRY		5.19.3
412	(@)-C	SIGNED BINARY SUBSTRACT WITH CARRY		5.19.3
413	(@)-CL	DOUBLE SIGNED BINARY SUBSTRACT WITH CARRY		5.19.3
414	(@)-B	BCD SUBSTRACT WITHOUT CARRY		5.19.4
415	(@)-BL	DOUBLE BCD SUBSTRACT WITHOUT CARRY		5.19.4
416	(@)-BC	BCD SUBSTRACT WITH CARRY		5.19.4
417	(@)-BCL	DOUBLE BCD SUBSTRACT WITH CARRY		5.19.4
420	(@)*	SIGNED BINARY MYLTIPLY		5.19.5
421	(@)*L	DOUBLE SIGNED BINARY MYLTIPLY		5.19.5
422	(@)*U	UNSIGNED BINARY MYLTIPLY		5.19.5
423	(@)*UL	DOUBLE UNSIGNED BINARY MYLTIPLY		5.19.5
424	(@)*B	BCD MYLTIPLY		5.19.6
425	(@)*BL	DOUBLE BCD MYLTIPLY		5.19.6

**Приложение В. Команды релейно-контактного программирования**

Код	Мнемоника	Наименование	Функция	Пункт
430	(@)/	SIGNED BINARY DIVIDE		5.19.7
431	(@)/L	DOUBLE SIGNED BINARY DIVIDE		5.19.7
432	(@)/U	UNSIGNED BINARY DIVIDE		5.19.7
433	(@)/UL	DOUBLE UNSIGNED BINARY DIVIDE		5.19.7
434	(@)/B	BCD DIVIDE		5.19.8
435	(@)/BL	DOUBLE BCD DIVIDE		5.19.8
480	(@)ADBL	DOUBLE BINARY ADD	Складывает два 8-разрядных двоичных чисел двойной длины (со знаком или без знака) и выдает результат в R и R+1.	5.21.5
481	(@)SBBL	DOUBLE BINARY SUBTRACT	Вычитает одно 8-разрядное двоичное число (со знаком или без) из другого и выдает результат в R+1 и R.	5.21.7
482	(@)MBSL	DOUBLE SIGNED BINARY MULTIPLY	Умножает два 32-битовых двоичных числа двойной длины со знаком и выдает 16-разрядный двоичный результат со знаком в R+3..R.	5.21.8
483	(@)DBSL	DOUBLE SIGNED BINARY DIVIDE	Делит одно 32-битовое двоичное число со знаком на другое и выдает 64-битовый двоичный результат со знаком в R+3..R	5.21.10
484	(@)MBS	SIGNED BINARY MULTIPLY	Умножает два 16-битовых двоичных числа со знаком и выдает 8-разрядный двоичный результат со знаком в R+1 и R.	5.21.7
485	(@)DBS	SIGNED BINARY DIVIDE	Делит одно 16-битовое двоичное число со знаком на другое и выдает 32-битовый двоичный результат со знаком в R+1 и R.	5.21.9
-	(@)XFR2	EM BLOCK TRANSFER	Переносит содержимое нескольких последовательных слов источника в несколько последовательных слов приемника. Для источника или приемника можно задавать слова в любом действующем банке EM. Команду можно использовать для изменения текущего банка.	5.16.11
-	(@)BXF2	EM BANK TRANSFER	Переносит содержимое нескольких последовательных слов источника в несколько последовательных слов приемника. Слова в текущем банке EM можно задать для источнике или приемника.	5.16.12
-	(@)IEMS	INDIRECT EM ADDRESSING	Изменяет адерата косвенной адресации DM (*DM) на DM или указанный банк EM. Данная команда может использоваться для изменения текущего банка EM.	5.26.15

## Приложение С

### Операции с флагами ошибок и арифметики

В таблице представлены команды, оказывающие воздействие на флаги N, OF, UF, ER, CY, GT, LT и EQ.

N указывает на отрицательный результат

OF указывает, что результат 16-битовых вычислений больше 32 767 (7FFF) или результат 32-битовых вычислений больше 2 147 483 647 (7FFF FFFF).

UF указывает, что результат 16-битовых вычислений меньше -32 768 (8000) или результат 32-битовых вычислений меньше -2 147 483 648 (8000 0000). (Подробности см. гл. 5).

ER указывает, что операнд находится вне допустимых значений.

CY указывает на результаты арифметических вычислений или сдвига.

GT указывает, что сравниваемая величина больше какого-то стандарта.

LT указывает, что сравниваемая величина меньше стандарта.

EQ указывает, что сравниваемая величина равна стандарту, а также то, что в арифметических вычислениях результат равен нулю. Подробности см. Гл. 5.

Стрелка в таблице показывает флаги, которые устанавливаются в 1 или 0 согласно результату исполнения команды.

Хотя команды TIM и CNT выполняются, когда ER = 1, другие команды, помеченные стрелкой в столбце ER, не выполняются, когда ER = 1. Все другие флаги в таблице не работают, когда ER = 1.

Команды, которых нет в таблице, на флаги влияния не оказывают. Хотя приведены только формы команд не фронта 0/1, версии команд фронта 0/1 воздействуют на флаги точно так же.

Все 8 флагов сбрасываются в 0 при исполнении команды END, так что их состояние нельзя индицировать с программатора.

Мнемоника	25503 (ER)	25504 (CY)	25505 (GR)	25506 (EQ)	25507 (LE)	25404 (OF)	25405 (UF)	25402 (N)
TIM	↕	-	-	-	-	-	-	-
CNT	↕	-	-	-	-	-	-	-
END	0	0	0	0	0	0	0	0
SFT	0	-	-	-	-	-	-	-
CNTR	↕	-	-	-	-	-	-	-
TIMH	↕	-	-	-	-	-	-	-
WSFT	↕	-	-	-	-	-	-	-
ASFT	↕	-	-	-	-	-	-	-
SCAN	↕	-	-	-	-	-	-	-
MCMP	↕	-	-	↕	-	-	-	-
CMP	↕	-	↕	↕	↕	-	-	-
MOV	↕	-	-	↕	-	-	-	↕
MVN	↕	-	-	↕	-	-	-	↕
BIN	↕	-	-	↕	-	-	-	0
BCD	↕	-	-	↕	-	-	-	-
ASL	↕	↕	-	↕	-	-	-	↕
ASR	↕	↕	-	↕	-	-	-	0
ROL	↕	↕	-	↕	-	-	-	-
ROR	↕	↕	-	↕	-	-	-	-

**Приложение С. Операции с флагами ошибок и арифметики**

Мнемоника	25503 (ER)	25504 (CY)	25505 (GR)	25506 (EQ)	25507 (LE)	25404 (OF)	25405 (UF)	25402 (N)
COM	⇕	-	-	⇕	-	-	-	-
ADD	⇕	⇕	-	⇕	-	-	-	-
SUB	⇕	⇕	-	⇕	-	-	-	-
MUL	⇕	-	-	⇕	-	-	-	-
DIV	⇕	-	-	⇕	-	-	-	-
ANDW	⇕	-	-	⇕	-	-	-	⇕
ORW	⇕	-	-	⇕	-	-	-	⇕
XORW	⇕	-	-	⇕	-	-	-	⇕
XNRW	⇕	-	-	⇕	-	-	-	⇕
INC	⇕	-	-	⇕	-	-	-	-
DEC	⇕	-	-	⇕	-	-	-	-
STC	-	1	-	-	-	-	-	-
CLC	-	-	-	-	-	-	-	-
MSG	⇕	-	-	-	-	-	-	-
LMSG	⇕	-	-	-	-	-	-	-
ADB	⇕	⇕	-	⇕	-	-	-	-
SBB	⇕	⇕	-	⇕	-	-	-	-
MLB	⇕	-	-	⇕	-	-	-	-
DVB	⇕	-	-	⇕	-	-	-	-
ADDL	⇕	⇕	-	⇕	-	-	-	-
SUBL	⇕	⇕	-	⇕	-	-	-	-
MULL	⇕	-	-	⇕	-	-	-	-
DIVL	⇕	-	-	⇕	-	-	-	-
BINL	⇕	-	-	⇕	-	-	-	0
BCDL	⇕	-	-	⇕	-	-	0	-
CMPL	⇕	-	⇕	⇕	⇕	-	-	-
MPRF	⇕	-	-	-	-	-	-	-
XFRB	⇕	-	-	-	-	-	-	-
LINE	⇕	-	-	⇕	-	-	-	-
COLM	⇕	-	-	⇕	-	-	-	-
SEC	⇕	-	-	⇕	-	-	-	-
HMS	⇕	-	-	⇕	-	-	-	-
BCNT	⇕	-	-	⇕	-	-	-	-
BCMP	⇕	-	-	-	-	-	-	-
APR	⇕	-	-	⇕	-	-	-	⇕
XFER	⇕	-	-	-	-	-	-	-
BSET	⇕	-	-	-	-	-	-	-
ROOT	⇕	-	-	⇕	-	-	-	-
XCHG	⇕	-	-	-	-	-	-	-
SLD	⇕	-	-	-	-	-	-	-
SRD	⇕	-	-	-	-	-	-	-
MLPX	⇕	-	-	-	-	-	-	-
DMPX	⇕	-	-	-	-	-	-	-
SDEC	⇕	-	-	-	-	-	-	-
FDIV	⇕	-	-	-	⇕	-	-	-
DIST	⇕	-	-	-	⇕	-	-	⇕

**Приложение С. Операции с флагами ошибок и арифметики**

<b>Мнемоника</b>	<b>25503 (ER)</b>	<b>25504 (CY)</b>	<b>25505 (GR)</b>	<b>25506 (EQ)</b>	<b>25507 (LE)</b>	<b>25404 (OF)</b>	<b>25405 (UF)</b>	<b>25402 (N)</b>
COLL	⇕	-	-	-	⇕	-	-	⇕
MOVB	⇕	-	-	-	-	-	-	-
MOVD	⇕	-	-	-	-	-	-	-
SFTR	⇕	⇕	-	-	-	-	-	-
TCMP	⇕	-	-	⇕	-	-	-	-
ASC	⇕	-	-	-	-	-	-	-
TTIM	⇕	-	-	-	-	-	-	-
ZCP	⇕	-	⇕	⇕	⇕	-	-	-
INT	⇕	-	-	-	-	-	-	-
SEND	⇕	-	-	-	-	-	-	-
SBS	⇕	-	-	-	-	-	-	-
SBN	⇕	⇕	⇕	⇕	⇕	⇕	⇕	⇕
WDT	⇕	-	-	-	-	-	-	-
IORF	⇕	-	-	-	-	-	-	-
RECV	⇕	-	-	-	-	-	-	-
MCRO	⇕	-	-	-	-	-	-	-
CPS	⇕	-	⇕	⇕	⇕	-	-	-
CPSL	⇕	-	⇕	⇕	⇕	-	-	-
ZCPL	⇕	-	⇕	⇕	⇕	-	-	-
NEG	⇕	-	-	⇕	-	-	⇕	⇕
NEGL	⇕	-	-	⇕	-	-	⇕	⇕
HEX	⇕	-	-	-	-	-	-	-
FCS	⇕	-	-	-	-	-	-	-
SRCH	⇕	-	-	⇕	-	-	-	-
MAX	⇕	-	-	⇕	-	-	-	⇕
MIN	⇕	-	-	⇕	-	-	-	⇕
SUM	⇕	-	-	⇕	-	-	-	⇕
PID	⇕	⇕	-	-	-	-	-	-
SCL	⇕	-	-	⇕	-	-	-	-
AVG	⇕	-	-	-	-	-	-	-
DSW	⇕	-	-	-	-	-	-	-
TKY	⇕	-	-	-	-	-	-	-
HKY	⇕	-	-	-	-	-	-	-
MTR	⇕	-	-	-	-	-	-	-
7SEG	⇕	-	-	-	-	-	-	-
IORD	⇕	-	-	⇕	-	-	-	-
IOWR	⇕	-	-	⇕	-	-	-	-
RXD	⇕	-	-	-	-	-	-	-
TXD	⇕	-	-	-	-	-	-	-
STUP	⇕	-	-	-	-	-	-	-
PMCR	⇕	-	-	-	-	-	-	-
CMCR	⇕	-	-	-	-	-	-	-
FPD	⇕	⇕	-	-	-	-	-	-
XDMR	⇕	-	-	-	-	-	-	-
EMBC	⇕	-	-	-	-	-	-	-
=	⇕	-	-	⇕	-	-	-	⇕

**Приложение С. Операции с флагами ошибок и арифметики**

<b>Мнемоника</b>	<b>25503 (ER)</b>	<b>25504 (CY)</b>	<b>25505 (GR)</b>	<b>25506 (EQ)</b>	<b>25507 (LE)</b>	<b>25404 (OF)</b>	<b>25405 (UF)</b>	<b>25402 (N)</b>
=L	⇕	-	-	⇕	-	-	-	⇕
=S	⇕	-	-	⇕	-	-	-	⇕
=SL	⇕	-	-	⇕	-	-	-	⇕
<>	⇕	-	-	⇕	-	-	-	⇕
<>L	⇕	-	-	⇕	-	-	-	⇕
<>S	⇕	-	-	⇕	-	-	-	⇕
<>SL	⇕	-	-	⇕	-	-	-	⇕
<	⇕	-	-	⇕	-	-	-	⇕
<L	⇕	-	-	⇕	-	-	-	⇕
<S	⇕	-	-	⇕	-	-	-	⇕
<SL	⇕	-	-	⇕	-	-	-	⇕
<=	⇕	-	-	⇕	-	-	-	⇕
<=L	⇕	-	-	⇕	-	-	-	⇕
<=S	⇕	-	-	⇕	-	-	-	⇕
>	⇕	-	-	⇕	-	-	-	⇕
>L	⇕	-	-	⇕	-	-	-	⇕
>S	⇕	-	-	⇕	-	-	-	⇕
>SL	⇕	-	-	⇕	-	-	-	⇕
>=	⇕	-	-	⇕	-	-	-	⇕
>=L	⇕	-	-	⇕	-	-	-	⇕
>=S	⇕	-	-	⇕	-	-	-	⇕
>=SL	⇕	-	-	⇕	-	-	-	⇕
TST	⇕	-	-	-	-	-	-	-
TSTN	⇕	-	-	-	-	-	-	-
+	⇕	⇕	-	⇕	-	⇕	⇕	⇕
+L	⇕	⇕	-	⇕	-	⇕	⇕	⇕
+C	⇕	⇕	-	⇕	-	⇕	⇕	⇕
+CL	⇕	⇕	-	⇕	-	⇕	⇕	⇕
+B	⇕	⇕	-	⇕	-	-	-	⇕
+BL	⇕	⇕	-	⇕	-	-	-	⇕
+BC	⇕	⇕	-	⇕	-	-	-	⇕
+BCL	⇕	⇕	-	⇕	-	-	-	⇕
-	⇕	⇕	-	⇕	-	⇕	⇕	⇕
-L	⇕	⇕	-	⇕	-	⇕	⇕	⇕
-C	⇕	⇕	-	⇕	-	⇕	⇕	⇕
-CL	⇕	⇕	-	⇕	-	⇕	⇕	⇕
-B	⇕	⇕	-	⇕	-	-	-	-
-BL	⇕	⇕	-	⇕	-	-	-	-
-BC	⇕	⇕	-	⇕	-	-	-	-
-BCL	⇕	⇕	-	⇕	-	-	-	-
*	⇕	-	-	⇕	-	-	-	⇕
*L	⇕	-	-	⇕	-	-	-	⇕
*U	⇕	-	-	⇕	-	-	-	⇕
*UL	⇕	-	-	⇕	-	-	-	⇕
*B	⇕	-	-	⇕	-	-	-	-
*BL	⇕	-	-	⇕	-	-	-	-



**Приложение С. Операции с флагами ошибок и арифметики**

Мнемоника	25503 (ER)	25504 (CY)	25505 (GR)	25506 (EQ)	25507 (LE)	25404 (OF)	25405 (UF)	25402 (N)
/	⇕	-	-	⇕	-	-	-	⇕
/L	⇕	-	-	⇕	-	-	-	⇕
/U	⇕	-	-	⇕	-	-	-	⇕
/UL	⇕	-	-	⇕	-	-	-	⇕
/B	⇕	-	-	⇕	-	-	-	-
/BL	⇕	-	-	⇕	-	-	-	-
ADBL	⇕	⇕	-	⇕	-	⇕	⇕	⇕
SBBL	⇕	⇕	-	⇕	-	⇕	⇕	⇕
MBSL	⇕	-	-	⇕	-	-	-	⇕
DBSL	⇕	-	-	⇕	-	-	-	⇕
MBS	⇕	-	-	⇕	-	-	-	⇕
DBS	⇕	-	-	⇕	-	-	-	⇕
XFR2	⇕	-	-	-	-	-	-	-
BXF2	⇕	-	-	-	-	-	-	-
IEMS	⇕	-	-	-	-	-	-	-



## Приложение D

### Бланки для записи распределения слов

В данном приложении содержатся бланки, которые может скопировать программист для записи распределения битов входов/выходов, рабочих битов, областей сохранения данных, таймеров и счетчиков.

#### Биты входов/выходов

Программист:

Программа:

Дата:

Страница:

Слово	Содержание	Примечание

#### Таймеры и счетчики

Программист:

Программа:

Дата:

Страница:

Слово	Содержание	Примечание



## Приложение Е

### Мнемокод программы

Наименование системы		Разработал	Проверил	Утвердил
Модель ПК	Лист N			

Адрес					Инструкция	Код	Операнды		
			0	0					
			0	1					
			0	2					
			0	3					
			0	4					
			0	5					
			0	6					
			0	7					
			0	8					
			0	9					
			1	0					
			1	1					
			1	2					
			1	3					
			1	4					
			1	5					
			1	6					
			1	7					
			1	8					
			1	9					



## Приложение Е

### Таблицы преобразования данных

#### Данные без знака

Десятичное	Двоично-десятичное	16-ричное	Двоичное
00	00000000	00	00000000
01	00000001	01	00000001
02	00000010	02	00000010
03	00000011	03	00000011
04	00000100	04	00000100
05	00000101	05	00000101
06	00000110	06	00000110
07	00000111	07	00000111
08	00001000	08	00001000
09	00001001	09	00001001
10	00010000	10	00001010
11	00010001	11	00001011
12	00010010	12	00001100
13	00010011	13	00001101
14	00010100	14	00001110
15	00010101	15	00001111
16	00010110	16	00010000
17	00010111	17	00010001
18	00011000	18	00010010
19	00011001	19	00010011
20	00100000	20	00010100
21	00100001	21	00010101
22	00100010	22	00010110
23	00100011	23	00010111
24	00100100	24	00011000
25	00100101	25	00011001
26	00100110	26	00011010
27	00100111	27	00011011
28	00101000	28	00011100
29	00101001	29	00011101
30	00110000	30	00011110
31	00110001	31	00011111
32	00110010	32	00100000

Двоичные данные со знаком

Десятичные	16-ричные 16 бит	16-ричные 32 бита
2 147 483 647	—	7FFF FFFF
2 147 483 646	—	7FFF FFFE
.	.	.
.	.	.
.	.	.
32 768	—	0000 8000
32 767	7FFF	0000 7FFF
32 766	7FFE	0000 7FFE
.	.	.
.	.	.
.	.	.
2	0002	0000 0002
1	0001	0000 0001
0	0000	0000 0000
-1	FFFF	FFFF FFFF
-2	FFFE	FFFF FFFE
.	.	.
.	.	.
.	.	.
-32 767	8001	FFFF 8001
-32 768	8000	FFFF 8000
-32 769	—	FFFF 7FFF
.	.	.
.	.	.
.	.	.
-2 147 483 647	—	8000 0001
-2 147 483 648	—	8000 0000



# Приложение G

## Расширенные коды ASCII

Биты 0... 3		Биты 4... 7													
Двоич.		0000	0001	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
	16-рич.	0	1	2	3	4	5	6	7	A	B	C	D	E	F
0000	0	NUL	DLE	" "	0	60 \	p		0	60 \	p				
0001	1	SOH	DC <sub>1</sub>	!	1	A	Q	a	q	!	1	A	Q	a	q
0010	2	STX	DC <sub>2</sub>	"	2	B	R	b	r	"	2	B	R	b	r
0011	3	ETX	DC <sub>3</sub>	#	3	C	S	c	s	#	3	C	S	c	s
0100	4	EOT	DC <sub>4</sub>	\$	4	D	T	d	t	\$	4	D	T	d	t
0101	5	ENQ	NAK	%	5	E	U	e	u	%	5	E	U	e	u
0110	6	ACK	SYN	&	6	F	V	f	v	&	6	F	V	f	v
0111	7	BEL	ETB	'	7	G	W	g	w	'	7	G	W	g	w
1000	8	BS	CAN	(	8	H	X	h	x	(	8	H	X	h	x
1001	9	HT	EM	)	9	I	Y	i	y	)	9	I	Y	i	y
1010	A	LF	SUB	*	:	J	Z	j	z	*	:	J	Z	j	z
1011	B	VT	ESC	+	;	K	[	k	{	+	;	K	[	k	{
1100	C	FF	FS	,	cell L		l	l	,	cell L		l	l		
1101	D	CR	GS	—	=	M	]	m	}	—	=	M	]	m	}
1110	E	S0	RS	.		N	^	n	_	.		N	^	n	
1111	F	S1	US	/	?	O	_	o	_	/	?	O	_	o	_



## Глоссарий

### **\*DM**

См. Косвенный адрес (в области DM).

### **AND**

Логическая операция над условиями, при которой результат является истиной, если только оба условия истины. В релейно-контактном программировании условия - это обычно состояния бит 1/0 или логическая комбинация таких состояний, называемая условиями исполнения.

### **APF (All Plastic fiber-optic cable)**

Полностью пластмассовые оптоволоконные кабели

### **AR область**

Область памяти, выделенная для флагов и битов управления.

### **ASCII**

American Standardcode For Information Interchange. Служит для кодирования символов при выдаче на принтер и другие внешние устройства.

### **AUTOEXEC.BAT**

Файл MS DOS, содержащий команды, автоматически исполняемые при запуске.

### **CH**

В сообщениях ПК обозначает слово. См. Слово.

### **CONFIG.SYS**

Файл MS DOS, содержащий параметры среды персонального компьютера.

### **CTS**

Clear-to-send (сброс передатчика) - сигнал, используемый при связи между устройствами, который указывает, что приемник готов принимать данные.

### **CU**

См. Флаг переноса

### **DIN профиль**

Профиль, который входит в монтажные канавки различных устройств, что позволяет на нем быстро и надежно монтировать устройства.

### **DIP-переключатель**

Dual-in-line package - корпус с двухрядным расположением (штырьковых) выводов, который монтируется на плате и служит для установки параметров работы.

### **DM область**

Область данных (Data memory) служащая для хранения только слов данных. Слова в области DM недоступны битами.

### **DM слово**

Слово в области DM

### **EEPROM**

ЭСПЗУ Электрически стираемое ПЗУ. Тип памяти, данные в которой можно стереть и переписать. Это делается через специальные входы микросхемы EEPROM и без удаления микросхемы с устройства, в котором она установлена.

### **EPROM**

СПЗУ Стираемое ПЗУ. Тип памяти, данные в которой можно стереть ультрафиолетом или другими способами, и переписать.

### **FA**

Factory automation - производственная автоматизация.

**FAL - ошибка**

Признак ошибки, вызванный из программы при исполнении команды FAL.

**FALS - ошибка**

Признак ошибки, вызванный из программы при исполнении команды FALS или вызванный системой.

**FCS**

См. КСК

**FCS**

(Frame checksum) - Контрольная сумма конверта. Результат выполнения операции ИСКЛЮЧАЮЩЕЕ ИЛИ с заданной зоной данных. КСК можно подсчитать на передающем и приемном устройствах при передаче данных, чтобы убедиться, что данные переданы правильно.

**HEX**

16-ричное представление числа Система представления чисел, на базе 16 цифр. В ПК все данные хранятся в двоичной форме, однако ввод и индикация на программирующих устройствах часто производятся в 16-ричном виде для упрощения операций. Каждая группа из четырех двоичных битов эквивалентна одной 16-ричной цифре.

**HR область**

Область памяти, которая сохраняет данные при отключенном питании и используется как область рабочих бит.

**I/O**

(Input/output) - входы/выходы

**IBM PC/AT или совместимый**

Компьютер, имеющий аналогичную структуру с IBM PC/AT, на котором работают программы IBM PC/AT.

**IN**

(Input) - вход

**INTERLOCK**

Метод программирования, служащий для того, чтобы работать с несколькими командами, как с группой, чтобы можно было сбросить всю группу вместе, когда не требуются индивидуальные условия. Сблокированная секция нормально выполняется для условия исполнения 1 и сбрасывается при условии исполнения 0.

**IOM**

(Input/Output Memory) - Память входов/выходов.

Включает области IR, SR, LR, HR, AR, текущие значения таймеров и счетчиков, DM 0000... DM 6143, EM 0000... EM 6143 (2.4).

**JIS**

(Japanese Industrial Standards) - Японские промышленные стандарты.

**LINK 1:1**

Связь между двумя ПК, организованная таким образом, чтобы образовать общую область LR.

**LR область**

Область данных, используемая при связи.

**LSS**

(Ladder support software) пакет программ, установленный на IBM PC/AT - совместимом компьютере для работы в качестве программирующего устройства.

**MONITOR**

Режим работы контроллера - отработка программы с возможностью вмешаться в ход отработки (принудительное включения/выключение входов/выходов). Служит для наблюдения или отладки ПК.

**OFF**

Состояние входа или выхода, когда сигнал отсутствует. Обычно говорят об уровне сигнала OFF (выключено) при низком входном напряжении, либо при непроводящем состоянии, но могут быть и противоположные случаи.

**ON**

Состояние входа или выхода, когда сигнал присутствует. Обычно говорят об уровне сигнала ON (включено) при высоком входном напряжении, либо при проводящем состоянии, но могут быть и противоположные случаи.

**OR**

Логическая операция над условиями, при которой результат является истиной, если хотя бы одно или два условия истинны. В релейно-контактном программировании условия - это обычно состояния битов (1 или 0) или логическая комбинация таких состояний, называемых условиями исполнения.

**OUT**

(Output) - выход

**PROGRAM**

Режим работы ПК - программирование, позволяющий ввод и отладку программ, но не позволяющий работу программы.

**PROM**

Programmable read-only memory: ППЗУ Программируемое постоянное запоминающее устройство. Тип ПЗУ, в котором программа или данные могут быть записаны пользователем, но потом эти данные сохраняются.

**PV**

(PRESENT VALUE) см. текущее значение

**PCF**

Акроним Plastic-clad optical fiber cable (оптоволоконный кабель в пластмассовой оболочке)

**RAM**

Read only memory - оперативное запоминающее устройство (ОЗУ). Не сохраняет данные при отключении питания.

**ROM**

Read-only memory: ПЗУ Постоянное запоминающее устройство. Тип ПЗУ, в которое писать нельзя. Микросхема ROM выпускается в уже готовыми данными или программой и их нельзя изменить. Однако программу или данные можно читать сколько угодно раз.

**RUN**

Режим работы контроллера - рабочий режим, работа по уже отлаженной программе.

SR область

Область памяти, которая содержит флаги и другие биты/слова со специальными функциями.

**SSS**

(SYSMAC support software) пакет программ, установленный на IBM PC/AT - совместимом компьютере для работы в качестве программирующего устройства.

**SV**

(SET VALUE) См. заданное значение

**TR бит**

Бит в области TR.

**TR область**

Область памяти, служащая для хранения условий исполнения, так что их можно будет загрузить позже для использования другими командами.

**UM область**

Область памяти, в которой хранится программа пользователя, т. Е. Программа, которая исполняется в настоящее время.

**адрес**

Число, служащее для указания места параметра или команды в памяти.

**адрес бита**

Место в памяти, где находится бит данных. Адрес бита содержит название области памяти, адрес слова и номер бита внутри слова.

**адрес слова**

Место в памяти, где хранятся слова. Адрес слова должен указывать (иногда берется по умолчанию) область данных и номер слова, к которому производится адресация.

**арифметический сдвиг**

Операция сдвига, при которой флаг переноса включается в операцию сдвига.

**арифметический сдвиг**

Операция сдвига, при которой флаг переноса включается в операцию сдвига.

**базовые команды**

Основные команды релейно-контактной схемы (в отличие от дополнительных команд)

**байт**

Единица данных, равная 8 битам, или половине слова.

**бит**

Самая малая единица информации, которую можно представить в вычислительном устройстве. Бит может иметь значение либо 1, либо 0, соответствующее электрическим сигналам ВКЛ или ВЫКЛ. Бит представляется одной двоичной цифрой. Некоторые биты ПК отведены для специальных целей, таких, как сохранение состояния входов от внешних устройств, а некоторые можно использовать для общего назначения в программировании.

**бит входа/выхода**

Бит в памяти ПК, использующийся для сохранения состояния входа/выхода. Биты входа отражают состояние реальных входов (входных зажимов).

Биты выхода содержат состояние для посылки на реальные входы (выходные зажимы).

**бит для разового включения**

Бит, включаемый в 1 или 0 на заданный интервал времени, больше, чем 1 цикл.

**бит перезапуска**

Бит, используемый для перезапуска ПК.

**бит самоподдержки**

Бит, который запрограммирован для поддержания состояния либо 0 либо 1 до установки или сброса с помощью заданных условий.

**битовый операнд**

Бит, заданный как операнд для команды.

**блок**

См. Логический блок и блок команд

**блок**

В терминологии OMRON, сборочная единица ПК.

**блок ASCII**

Интеллектуальный блок входов/выходов для программирования на BASIC. При подключению к NSU системы NET LINK запросы можно посылать на другие узлы.

**блок входа/выхода**

Блок ПК, физически подключенные к входным/выходным устройствам для ввода и выдачи сигналов. Включают блоки входа и блоки выхода, несколько модификаций входных и выходных.

**блок команд**

Группа команд, логически связанные на релейно-контактной схеме. Логический блок включает все командные линии, соединяющиеся друг с другом, от одной или более линий подключенных к левой шине, к одной или более выходных (правых ) команд, подключенных к выходной (правой) шине.

**блок питания**

Блок, подключенный к ПК, обеспечивающий напряжение, требуемой другими блоками.

**блочный ПК**

ПК, составленный из отдельных компонентов, или блоков. У данных ПК отдельный блок не идентифицируется как ПК. ПК является функциональным объединением блоком

**вложенность**

Программирование одного цикла в другом, программирование вызова подпрограммы из самой подпрограммы, или программирование одного перехода внутри другого.

**возврат**

Процесс, когда исполнение команд возвращается из подпрограммы в главную программу (обычно в точку, из которой был вызов подпрограммы).

**возвращение**

Процессы копирования данных либо с внешнего устройства, или из области хранения в активную область системы, такую как буфер дисплея.

**возобновление**

Процесс, в котором устройство будет снова пытаться передавать данные после получении ошибки из приемника.

**время исполнения**

Время, требующееся для ЦПУ для выполнения либо отдельной команды, либо всей программы.

**время исполнения команды**

Время, требуемое для исполнения команды. Время для любой команды может варьировать от условий исполнения и используемых операндов.

**время контроля за ответом**

Время, которое устройство ожидает при пересылке данных до тех пор, пока не будет уверенности, что произошел сбой.

**время реакции на вход**

Время, требующееся для выдачи выходного сигнала с ПК в ответ на входной сигнал, полученный с внешнего устройства.

**время сканирования**

См. Время цикла

**время цикла**

Время необходимое для осуществления одного цикла ПК.

**вход**

Сигнал, приходящий с внешнего устройства на ПК. Термин часто используется как общий термин для обозначения входных сигналов.

**вход уменьшения счетчика**

Сигнал входа, служащий для декрементирования счетчика при смене сигнала с 0 на 1.

**входное устройство**

Внешнее устройство, посылающее сигнал в ПК.

**входное/выходное устройство**

Устройство, подключенное к входным или выходным разъемам ПК. Может быть либо частью управляющей системы, либо функционируют для помощи в управлении другими устройствами, либо частью управляемой системы

**входной бит**

Бит области IR, выделенный для хранения состояния входа.

**входной сигнал**

Изменение состояния входа ПК. Обычно говорят об активном сигнале, когда состояние входа изменяется с низкого на высокое напряжения, либо из непроводящего в проводящее состояние.

**входной сигнал сдвига**

Входной сигнал, переход которого из 0 в 1 вызывает сдвиг данных на 1 бит.

**вызов (подпрограммы)**

Процесс, при котором исполнение команд переходит из главной программы к подпрограмме. Подпрограмму можно вызвать либо из программы, либо по прерыванию.

**выход**

Сигнал, посылаемый из ПК на внешнее устройство. Термин часто используется как общий термин для обозначения выходных сигналов.

**выходное устройство**

Внешнее устройство, принимающее сигнал от ПК.

**выходной бит**

Бит области IR, выделенный для хранения состояния, которое должно быть послано на выходное устройство.

**выходной сигнал**

Сигнал, посылаемый на выходное устройство. Обычно говорят о том, что выходной сигнал существует, когда состояние точки выхода изменяется с низкого на высокое напряжения, либо из непроводящего в проводящее состояние.

**выходные команды**

Команды, помещаемые с правой стороны на релейно-контактной схеме и использующие последние условия исполнения в командной линии.

**главная программа**

Вся программа, за исключением подпрограмм и программ прерываний.

**граница области данных**

Самый больший адрес в области данных Назначая операнд, требующий нескольких слов, необходимо убедиться, что не превышен верхний адрес области данных.

**двоично-десятичное вычисление**

Математическое вычисление, использующее числа, представленные в двоично-десятичном виде.

**двоично-десятичное представление числа**

Система представления чисел, когда четыре двоичных бита эквивалентны десятичной цифре.

**двоичное вычисление**

Математическое вычисление, использующее числа, представленные в двоичном виде.

**двоичное представление числа**

Система представления чисел, когда числа записываются только цифрами 0 и 1. Каждая группа из четырех двоичных бит эквивалентна одной 16-ричной цифре. Двоичные данные в памяти для удобства выражают в 16-ричном виде.

**двоичное число без знака**

Двоичное значение, загруженное в память без указания знака.



**двоичное число со знаком**

Двоичное значение, хранящееся в памяти, с битом, который указывает на знак (+ или -).

**декремент**

Уменьшение (обычно на 1)

**десятичное представление числа**

Система счисления, в которой числа представляются на базе 10. В ПК все данные в конце концов хранятся в двоичной форме, 4 двоичных бита часто используются для представления одной десятичной цифры (двоично-десятичная система).

**десятичное число с плавающей точкой**

Десятичное число, представленное как число (мантисса), умноженное на 10 в степени, напр.  $0.538 \times 10^{-5}$ .

**диск данных**

Дискета, служащая для сохранения программы пользователя, содержания области DM и других данных пользователя.

**дискрета**

Единица измерения

**дистанция передачи**

Расстояние, на которое можно передать сигнал.

**длина данных**

В связи, количество бит, которые будут восприниматься как единый блок при передаче данных.

**загрузка**

Процесс копирования данных либо с внешних устройств, либо с области сохранения, в активную часть системы, такую, как буфер дисплея.

**загрузка на верхний уровень**

Процесс передачи программы или данных с низшего уровня или ведомого компьютера на верхний уровень или ведущий компьютер. Если работает программирующее устройство, оно считается управляющим компьютером.

**загрузка с верхнего уровня**

Процесс передачи программы или данных компьютера верхнего уровня, или ведущего компьютера, на ведомый компьютер. При работе с программирующим устройством, оно считается ведущим компьютером.

**заданное значение**

Значение, от которого декрементирующий счетчик начинает отсчет или до которого досчитывает инкрементирующий счетчик (т. Е. Максимальное число) или время, от которого или до которого таймер начинает отсчет.

**задержка OFF (0)**

Задержка от момента, когда сигнал на передающей стороне выключился в 0 (например, на входном устройстве или ПК) и моментом, когда сигнал воспримется как 0 на принимающей стороне (например, на выходном устройстве или ПК).

**задержка ON (1)**

Задержка от момента, когда сигнал на передающей стороне включился в 1 (например, на входном устройстве или ПК) и моментом, когда сигнал воспримется как 1 на принимающей стороне (например, на выходном устройстве или ПК).

**задержка включения входов/выходов**

Либо задержка от посылки сигнала на выход до включения реального выхода  
Либо задержка от изменения состояния на реальном входе до того, как сигнал изменении будет воспринят в ПК.

**замаскированный бит**

Бит, состояние которого временно становится неэффективным.

**запрограммированное предупреждение**

Предупреждение, выдаваемое в результате выполнения команды, предназначенной для вызова предупреждения, в отличие от предупреждений, выдаваемых системой.

**запрограммированный признак ошибки**

Признак ошибки, выдаваемый в результате выполнения команды, предназначенной для вызова признака ошибки из программы, в отличие от признака ошибки, выдаваемого системой.

**зарезервированное слово**

Слово в памяти, зарезервированное для специальных задач, и недоступный для пользователя.

**зарезервированный бит**

Бит, недоступный для пользователя.

**защита от записи**

Состояние, в котором содержимое устройства хранения информации нельзя изменить.

**импульс управления**

Сигнал, который подсчитывает счетчик.

**инверсное условие**

См. Нормально закрытое условие.

**инициализация**

Часть процесса пуска, в котором очищается часть адресов памяти. проверяются установочные параметры системы и устанавливаются значения по умолчанию.

**инкремент**

Увеличение на 1

**инкрементирующий вход счетчика**

Входной сигнал, служащий для увеличения счетчика на 1 при изменении сигнала с 0 на 1.

**инсталляция**

Подготовка, необходимая для использования программы или пакета программ, таких как LSS или SSS. на компьютере.

**интерфейс**

Стык между системами или устройствами и обычно включает изменение в способе представления данных. Устройства интерфейса выполняют такие операции, как кодирование, форматирование данных.

**интерфейс RS-232C**

Промышленный стандартный интерфейс для связи.

**интерфейс с управляющим компьютером**

Интерфейс, позволяющий осуществлять связь с управляющим компьютером.

**исключающее ИЛИ**

Логическая операция над условиями, при которой результат является истиной, если истинно одно, и только одно условие. В релейно-контактном программировании условия - это обычно состояния битов (1 или 0) или логическая комбинация таких состояний, называемых условиями исполнения.

**исключающее ИЛИ НЕ**

Логическая операция над условиями, при которой результат является истиной, если только оба условия истинны или оба условия ложны. В релейно-контактном программировании условия - это обычно состояния битов (1 или 0) или логическая комбинация таких состояний, называемых условиями исполнения.

**источник**

Ячейка (ячейки) памяти, откуда команда берет данные, над которыми она совершает действие, в отличие от ячеек, куда помещаются данные. Ячейка (ячейки) памяти, куда помещаются результаты, называется приемник.

**кабель связи**

Кабель для передачи данных между участками системы, отвечающий стандартам RS-232C или RS-422.

**кадр**

Блок данных для приема/передачи, имеющий начало (символ @), код заголовка, информацию, контрольную сумму и оканчивается кодом окончания (терминатор - символы \* \_)

**код имени команды**

Код имени команды, который указывает, какую команду выполнять

**код ответа**

Код, посылаемый в ответ на передачу данных, который указывает, как прошла передача данных.

**код ошибки**

Цифровой код, выданный для индикации того, что произошла ошибка, и некоторая информация о природе ошибки. Некоторые ошибки генерируются системой; другие задаются в программе оператором.

**код символа**

Числовой (обычно двоичный) код, служащий для представления буквенно-цифрового символа.

**команда пересылки данных**

Команда для пересылки данных из одного места памяти в другое. Данные в источнике не изменяются.

**команда фронта 0/1**

Команда, которая выполняется только один раз, когда условие исполнения на входе изменяется с 0 на 1. Команда, срабатывающая не по фронту, выполняется в каждом цикле, пока условие исполнения на входе = 1.

**команда включения на 1 цикл**

Команда, включающая бит операнда в 1 только на 1 цикл, когда условие исполнения на входе изменяется либо с 0 на 1 (DIFU) или с 1 на 0 (DIFD).

**команда логического блока**

Команда для логического объединения условий-результатов логического блока с текущим условием. Текущее условие может быть либо результатом одного условия, либо другого логического блока. Командами логического блока являются команды and load и or load.

**команда сравнения**

Команда, служащая для сравнения данных в различных местах памяти для определения соотношения между этими данными.

**команда, управляющая битами**

Команда, которая управляет состоянием отдельного бита (в отличие от команды, управляющей состоянием целого слова).

**командная линия**

Группа условий, лежащих на одной горизонтальной линии релейно-контактной схемы. Командные линии могут разветвляться или соединяться для образования блоков команд. Также называется ступенькой.

**константа**

Операнда, в котором указана сама величина (в отличие от адреса). Обозначается символом #.

**контроль на четность - нечетный**

Настройка при связи, когда число бит в состоянии 1 всегда будет нечетным.

**контроль на четность - четный**

Настройка при связи, когда число бит в состоянии 1 всегда будет четным.

**контроль четности**

Проверка на четность для определения, не испорчены ли данные при передаче.

**контрольная сумма**

Сумма, которая передается при связи вместе с пакетом данных. КС пересчитывается для принимаемых данных для того, чтобы убедиться, что полученные данные не запорчены.

**контрольный таймер**

Таймер в системе, который обеспечивает чтобы время сканирования оставалось в заданных рамках. При переходе за эти границы либо выдается либо предупреждение, либо по достижении определенного значения останавливается работа ПК.

**конфигурация ПК**

Совокупность блоков ПК, объединенных в одно устройство.

**конфигурация системы**

Порядок, в котором соединены блоки. Данный термин относится ко всем блокам и устройствам.

**косвенный адрес**

Адрес, содержимое которого указывает на адрес. Содержимое второго адреса будет использовано как операнд.

**КСК**

(Frame checksum) - Контрольная сумма конверта. Результат выполнения операции ИСКЛЮЧАЮЩЕЕ ИЛИ с заданной зоной данных. КСК можно подсчитать на передающем и приемном устройствах при передаче данных, чтобы убедиться, что данные переданы правильно.

**линия связи управляющим компьютером**

Линия связи, связывающая ПК с управляющим компьютером для возможности наблюдения или управления с управляющего компьютера.

**логическая команда**

Команда, служащая для логического объединения содержания двух слов и выдачи логического результата в заданное слово результата. Логические команды объединяют все биты с одинаковыми номерами в двух словах и выдают результат в бит с тем же номером в заданном слове результата.

**логический блок**

Группа команд, логически связанных на релейно-контактной схеме, и требующих блоковых команд для связи их с другими командами или логическими блоками.

**маскирование**

“Закрытие” сигнала прерывания так, что вызов прерывания не действует до удаления маски.

**мегабайт**

Единица хранения информации, равная миллиону бит.

**мерцающий бит**

Бит, который запрограммирован на включение в 1 и 0 с заданной частотой.

**младший бит(слово)**

Младший из группы битов, обычно во всем слове, или группы слов.

**мнемокод**

Форма релейно-контактного программирования, который состоит в списке команд без использования схемы.

**мощность переключения**

Максимальные напряжение/ток, которые реле может переключать в 1 и 0.

**настройка четности**

Настройка количества бит в состоянии 1 в слове или другой единице данных, так что общее количество всегда является либо четным, либо нечетным числом. Контроль на четность обычно используется для проверки правильности данных после передачи для определения, осталось ли число битов в состоянии 1 четным или нечетным.

**НЕ**

Логическая операция, инвертирующая состояние операнда. Например, AND NOT указывает, на операцию И с состоянием бита, противоположным текущему.

**нефатальная ошибка**

Ошибка аппаратной или программной части, вызывающая предупредительное сообщение, но не останавливающая работу ПК.

**номер бита**

Число, указывающее положение бита в слове. Бит 00 - самый правый (младший) бит; бит 15 - самый левый (старший) бит.

**номер блока**

Номер, приписанный каждому блоку, для облегчения его идентификации при обращении.

**номер перехода**

Определитель, используемый с командой перехода, для указания точек, с которой и на которую должен быть переход.

**номер прерывания**

Определитель, служащий для идентификации подпрограммы, чтобы можно было осуществить вызов подпрограммы.

**номер сообщения**

Номер сообщения, вызванного командой MESSAGE.

**нормально закрытый вход**

Вход, который нормально закрыт, т. Е. Входной сигнал считается активным, когда входная цепь разомкнута.

**нормально открытый вход**

Вход, который нормально открыт, т. Е. Входной сигнал считается активным, когда входная цепь замкнута.

**область**

Секция в памяти, имеющая свое имя и буквенное обозначение. Есть область данных и область памяти.

**область протокола ошибок**

Область, служащая для хранения записей, в которой указаны время и причина ошибок, которые произошли в системе.

**область связь по линии данных**

Область общих данных, связанная с линией данных.

**область данных**

Область в памяти ПК, выделенная для хранения заданного типа данных.

**область только для чтения**

Область памяти, откуда пользователь может только читать состояние бит, но не может туда писать

**обновление**

Процесс обновления состояния выходов на внешние устройства, чтобы они соответствовали битам выхода в памяти, и обновление входных битов в памяти, чтобы они соответствовали состоянию реальных входов.

**обновление входов/выходов**

Процесс обновления состояния выходов на внешние устройства, чтобы они соответствовали битам выхода в памяти, и обновление входных битов в памяти, чтобы они соответствовали состоянию реальных входов.

**обработка события**

Обработка, вызванная как реакция на событие, например, на прерывание.

**обслуживание**

Процесс, при котором ПК проверяет коннектор или блок, чтобы определить, не требуется ли обслуживание.

**обслуживание периферии**

Обслуживание сигналов с и на периферийные устройства, включая обновление, связь и т. д.

**общая диспетчеризация**

Часть времени цикла ПК, включающая отработку общих задач, требуемых при работе ПК.

**общее число входов/выходов**

Общее число входов/выходов, которыми может управлять ПК. Данное число меняется от около ста для малых ПК до 2 тысяч для самых больших.

**общие данные**

Данные в памяти ПК, к которым имеет доступ другой ПК из одной системы. У каждого ПК выделена специальная область памяти. Каждой ПК записывает в выделенную секцию и читает из секции, выделенной другим ПК, с которым у него общие данные.

**он-лайн**

Режим работы компьютера по линии связи с другим устройствами

**онлайновое редактирование**

Процесс коррекции программы в ПК прямо с программирующего устройства. Возможно только в режимах PROGRAM или MONITOR. В режиме MONITOR программу можно менять во время исполнения.

**операнд**

Значения, приписанные как данные для команды. Операнд может быть либо константой, представляющей числовое значение, которое будет использоваться командой, либо адресом, в котором находятся данные, которые будут использоваться.

**описатель бита**

Операнд, служащий для описания бита в слове, который будет использован в качестве операнда команды.

**описатель цифры**

Операнд, использующийся для назначения цифры или цифр слова, которые используются командой.

**определитель**

Число, служащее как операнд для команд, но которое служит только для определения самой команды, а не данных, которыми оперирует команда. Определители включают номера переходов, номера подпрограмм и т. д.

**отладка**

Процесс, в котором черновой вариант программы корректируется до тех пор, пока программа не заработает как запланировано. Отладка включает устранение синтаксических ошибок и точную настройку времен и координацию операций управления.

**отрицательная задержка**

Задержка, установления для трассировки данных, при которой регистрация данных начинается на заданное время перед сигналом трассировки.

**оф-лайн**

Режим работы компьютера автономно без связи с другим устройствами

**ошибка аппаратной части**

Неисправность, причина которой аппаратной части электрических компонентов ПК, в отличие от ошибок программной части, причина которых - в программа.

**ошибка инициализации**

Ошибка, произошедшая в аппаратной либо программной части при пуске ПК.

**ошибка синтаксиса**

Ошибка в написании программы. Ошибки могут включать

**Ошибки правописания (т. Е. Несуществующий функциональные код )**

Ошибки в задании операндов при допустимых параметрах (например, задать для записи биты, в которых возможно только чтение)

Ошибки в применении команд (например, вызов несуществующей подпрограммы).

**ошибка системной программы**

Ошибка, вызванная системной программой.

**память трассировки**

Область памяти, служащая для хранения результатов операции трассировки.

**параметр управления**

Операнд, в котором задано, как выполнять команду. Параметр управления может указывать часть слова, которая используется в качестве операнда, может определять приемник для команды пересылки данных, может указывать размер таблицы данных, используемых в команде и т.д.

**передача**

Процесс передачи данных из одного места в другое в пределах ПК, или между ПК и внешними устройствами. При передаче данных, посылается обычно копия данных, т. Е. Содержимое источника остается неизменным.

**передача данных**

Передача данных из одного места в памяти в другое, либо в пределах одного устройства, либо между разными устройствами, соединенными линией связи или сетью.

**передний фронт импульса**

Точка, где сигнал изменяется с 0 на 1

**переключатель защиты от записи**

Переключатель, служащий для защиты от записи содержимого устройства хранения информации, например, гибкий диск. Если отверстие в левом верхнем углу открыто, информация на нем нельзя изменить.

**переписать**

Изменить содержание ячеек памяти с потерей прошлого состояния.

**переполнение**

Состояние, при котором превышена емкость памяти, отведенная под хранение данных.

**переход**

Тип программирования, когда исполнение переходит с одной точки программы к другому без выполнения команд между ними.

**периферийное устройство**

Устройство, подключенное к ПК для помощи в работе системы. Периферийные устройства включают принтеры, программирующие устройства, внешние устройства памяти и т. Д.

**ПК**

Программируемый контроллер

**по умолчанию**

Значение, автоматически задаваемое ПК, когда пользователь не указывает другое значение. Большинство устройств принимает значение по умолчанию при включении питания.

**подпрограмма**

Группа команд, расположенных отдельно от главной программы и исполняемая только при вызове из главной программы или из прерывания.

**подсказка**

Сообщение или символ, который появляется на дисплее для того, чтобы потребовать ввода от оператора.

**положительная задержка**

Задержка, установленная для трассировки данных, при которой регистрация данных начинается на заданное время после сигнала трассировки.

**порт**

Разъем на ПК или компьютере, служащий для связи с внешними устройствами.

**последовательное подключение**

Соединение, при котором блоки связаны в цепь.

**правая команда**

Команда, расположенная на релейно-контактной схеме справа

**прерывание (сигнал)**

Сигнал, останавливающий нормальное исполнение программы и вызывающий подпрограмму или другие процессы обработки.

**прерывание входов/выходов**

Прерывание, вызванное сигналом со входов/выходов.

**прерывание по расписанию**

Прерывания, которые вызываются автоматически системой через заданные интервалы времени или из места программы, указанного оператором. Прерывание по расписанию вызывают выполнение заданной подпрограммы, которую можно использовать для выполнения команд, которые должны выполняться через заданные интервалы времени.

**префикс области**

Префикс из одной или двух букв для обозначения области памяти в ПК. Все области памяти, за исключением областей IR и SR, при адресации требуют префикса.

**приемник**

Ячейка (ячейки) памяти, куда команда помещает данные, над которыми она совершила действие, в отличие от ячеек, откуда берутся данные. Ячейка (ячейки) памяти, откуда команда берет данные, над которыми она совершает действие, называется источник.

**Принудительная установка**

Процесс принудительной установки в 1 бита с программирующего устройства. Биты обычно устанавливаются в 1 как результат исполнения программы.

**Принудительный сброс**

Процесс принудительного сброса в 0 бита с программирующего устройства. Биты обычно устанавливаются в 0 как результат исполнения программы.

**программа прерывания**

Программа, выполняющаяся как реакция на прерывание.

**программатор**

Переносное программирующее устройство для ПК

**программируемый контроллер**

Вычислительное устройство, способное принимать входные сигналы с внешних устройств и выдавать выходные сигналы на внешнее устройство согласно программы,



загруженной в память. Пк используются для автоматизации внешних устройств. Хотя есть пк в виде единого устройства, чаще пк собираются из отдельных блоков.

**программирующее устройство**

Периферийное устройство, служащее для ввода программы в ПК или для изменения или контроля программы, уже загруженной в ПК. Есть специальные программирующие устройства, такие, как программатор, и неспециализированные, такие, как управляющий компьютер.

**программная защита**

Средство защиты данных (от возможных изменений) при помощи программных средств, в отличие от реальных ключей или других аппаратных средств.

**протокол**

Параметры и процедуры, стандартизированные для того, чтобы сделать возможным связь двух устройств, или сделать возможным связь оператора или программатора с устройством.

**прямой выход**

Метод, при котором результаты исполнения программы выдаются не на биты выхода, прямо на реальный выход для устранения влияния времени цикла.

**ПТ**

программируемый терминал

**рабочая область**

Часть памяти, содержащая рабочие слова/биты.

**рабочее слово**

Слово, которое можно использовать для вычисления данных или других операций, т. Е. "рабочее пространство" в памяти. Большая часть области *ir* всегда зарезервированы как рабочие слова. Часть других областей, не используемые для специальных целей, используются в качестве рабочих битов.

**рабочий бит**

Бит в рабочем слове

**размаскирование**

"Открытие" замаскированного сигнала прерывания так, что вызов прерывания действует и прерывание будет обрабатываться.

**размаскированный бит**

Бит, чье значение неэффективно. (см. Замаскированный бит)

**распределенное управление**

Концепция автоматизации, при которой участок системы автоматизации располагается рядом с управляемыми устройствами, т. Е. Управление децентрализуется и "распределяется" по системе. Распределенное управление - концепция, основанная на системе пк.

**расширенный счетчик**

Счетчик, созданный в программе с последовательным использованием двух и более команд счетчиков. Такой счетчик способен считать большие значения, чем позволяют стандартные команды.

**расширенный таймер**

Таймер, созданный в программе с последовательным использованием двух и более команд таймеров. Такой таймер способен отсчитывать большие интервалы времени, чем позволяют стандартные команды.

**реверсивный регистр сдвига**

Регистр сдвига, который может сдвигать данные в обоих направлениях в зависимости от заданных условий.

**реверсивный счетчик**

Счетчик, который может и увеличивать и уменьшать свое значение в зависимости от заданных условий.

**регистр сдвига**

Одно или более слов, в которых данные сдвигаются на заданное число дискрет вправо или влево с дискретами бит, цифра или слово. В регистре циклического сдвига данные выдвигаются из одного конца и вдвигаются в другой конец регистра. В других регистрах сдвига новые данные (указанные данные, нуль (нули) или единица (единицы)) вдвигаются в один конец, а выдвигаемые с другого конца теряются.

**регулярный импульс**

Импульс на специально отведенном выходе, служащий для организации таймерных операций. Имеются импульсы различной ширины и, следовательно, различной частоты.

**режимы работы**

Один из трех режимов работы ПК: RPROGRAM, MONITOR, RUN.

**резервная копия**

Копия существующих данных, которая сохранится в случае искажения или потери исходных данных

**релейно-контактная команда**

Команда в программе, указывающая ПК на действие, которое должно быть произведено, и на данные, которые нужно при этом использовать. Команды можно использовать для простой установки бит в 0 или 1, или для выполнения более сложных операций, таких как преобразование и/или передачи больших блоков данных.

**релейно-контактная команда**

Команда, которая представляет условия на релейно-контактной схеме. Другие команды располагаются вдоль правой шины релейно-контактной схемы и называются выходными командами.

**релейно-контактная схема**

Форма программирования, появившаяся из релейных систем управления, которая использует релейные символы для представления алгоритма работы программы. Внешний вид программы похож на лестницу, отсюда и другое название (лестничная диаграмма).

**релейное управление**

Предшественник ПК. При релейном управлении группы реле соединялись для образования цепей управления. В ПК это делается программированием.

**самодиагностика**

Процесс, при котором система сама проверяет свою работу и вызывает предупреждение или признак ошибки в случае ненормальной работы.

**сбой при работе**

Ошибка, которая происходит при работе ПК, в отличие от ошибок инициализации, которая появляется перед фактическим началом работы.

**сброс (RESET)**

Процесс установки бита или сигнала в 0 или изменения текущего значения таймера и счетчика в заданное значение или 0.

**связь по линии данных**

Операция автоматической передачи данных, что позволяет ПК или блокам ПК обмениваться данными с помощью области общих данных.

**сигнал управления**

Сигнал, посылаемый из ПК, для воздействия на операцию системы управления.

**символ релейно-контактной схемы**

Символ, используемый для изображения программы в виде релейно-контактной схемы.

**синтаксис**

Форма программы (в отличие от значения)

**синхронное исполнение**

Исполнение программ и операций обслуживания при котором данные работы синхронизированы, так что все операции обслуживания выполняются каждый раз при исполнении программы.

**система ПК**

ПК со всеми блоками, подключенными к внешним устройствам. Границами системы ПК являются:

Сверху: ПК и программа и ЦПУ

Снизу: блоки ПК

**система управления**

Все компоненты (аппаратные и программные), служащие для управления другими устройствами. Система управления включает ПК, программы ПК и устройства входа/выхода, которые используются для управления или обратной связи с управляемой системой.

**системная ошибка**

Ошибка, вызванная системой, в отличие от вызванной исполнением команды, предназначенной для вызова признака ошибки.

**сканирование**

Процесс исполнения релейно-контактной программы. Программа последовательно просматривается с начала до конца и каждая команда выполняется по очереди в зависимости от условий исполнения.

**скорость передачи данных**

Скорость передачи данных операции связи между двумя устройствами, единица измерения - бит/с

**словный операнд**

Слово, заданное как операнд для команды.

**слово**

Единица данных, состоящая из 16 бит. Все области данных состоят из слов. Некоторые области данных доступны только словами. Другие доступны только битами и словами.

**слово входа/выхода**

Слово в области IR, выделенное блоку входов/выходов и использующееся для хранения состояния блока входов/выходов.

**слово результата**

Слово, используемое для помещения результата от исполнения команды

**смещение**

Положительная или отрицательная величина, добавляемая к базе, такой как адрес, для указания на требуемое значение.

**создание общей области данных**

Процесс, при котором создаются области общих данных между двумя или более ПК.

**сообщение о системной ошибке**

Сообщение об ошибке, вызванной системой, в отличие от сообщения, вызванного исполнением команды, предназначенной для вызова сообщения.

**сохранение**

Процесс записи в память для постоянного хранения программы, которая написана в буфере дисплея.

**специальная команда**

Команда, вводимая своим функциональным кодом, в отличие от базовых команд, которые составляют основу релейно-контактной схемы.

**старший бит(слово)**

Старший из группы битов, обычно во всем слове, или группы слов.

**ступенька**

См. Командная линия

**счетчик**

Специально выделенная группа цифр или слов в памяти, служащая для подсчета количества, сколько раз произошел указанный процесс; или область в памяти, доступная битам таймера и счетчика и служащая для подсчета того, сколько раз условие исполнения изменилось с 0 на 1.

**таймер**

Ячейки памяти, доступные командами TIM/CNT и отсчитывающие время вниз от заданного значения. Таймеры включаются в 1 и сбрасываются в соответствии с условиями исполнения.

**текущее значение**

Текущее значение, зарегистрированное в устройстве при работе. Обычно применяется к таймерам и счетчикам.

**терминатор**

Код окончания запроса или ответа при передаче данных

**точка входа**

Место, в котором входной сигнал входит в ПК. Физически соответствуют клеммам или ножкам разъемов на блоке.

**точка входа/выхода**

Место, в котором входной сигнал входит в систему ПК, или в котором выходной сигнал выходит из системы ПК. Физически соответствуют клеммам или ножкам разъемов на блоке. С точки зрения программы, соответствуют битам входов/выходов в области IR.

**точка выхода**

Место, в котором выходной сигнал выходит из ПК. Физически соответствуют клеммам или ножкам разъемов на блоке.

**трассировка**

Операция, при которой исполняется программа и результаты сохраняются для последующего пошагового анализа и отладки.

**трассировка адреса**

Процесс, в котором при исполнении программы регистрируются изменения в указанном месте памяти.

**трассировка данных**

Процесс, когда при исполнении программы регистрируются изменения содержания указанных участков памяти.

**умолчание**

См. По умолчанию

**УППК**

Установочные параметры программируемого контроллера

**управляющий бит**

Бит в области памяти, устанавливаемый либо из программы, либо с программирующего устройства для целей управления, например, бит перезапуска включается в 1 и 0 для перезапуска блока.

**управляющий компьютер**

Компьютер, использующийся для передачи или приема данных с ПК в управляющей системе Host Link. Управляющий компьютер служит для управления данными и управления системой. В качестве управляющего компьютера обычно используется персональный компьютер или бизнес-компьютер.

**условие**

Символ на командной линии релейно-контактной схеме, которая управляет выходной командой. Каждому условию приписан в памяти бит, который определяет его состояние. Состояние бита определяет следующее условие исполнения. Условия соответствуют командам LOAD, LOAD NOT, AND, AND NOT, OR или OR NOT.

**условие исполнения**

Состояние 1 или 0, при которых исполняется команда. Условие исполнения задается логической комбинацией условий на одной командной линии релейно-контактной схемы (до исполняемой команды).

**установка**

Процесс установки бита или сигнала в 1.

**установочные параметры ПК**

Группа параметров, управляющих работой ПК, задаваемых с программирующего устройства,

**установочные параметры системы**

Установочные параметры системы для программирующего устройства, например, LSS или SSS.

**фатальная ошибка**

Ошибка, вызывающая остановку работы ПК и требующая исправления перед продолжением работы

**флаг**

Специальный бит в памяти, устанавливаемый системой для индикации состояния определенного типа операций. Некоторые флаги, такие, как перенос, может установить оператор, а также можно устанавливать из программы.

**флаг завершения**

Флаг, используемый таймерами и счетчиками, который устанавливается в 1, когда таймер отсчитал заданное время или счетчик отсчитал заданное значение.

**флаг переноса**

Флаг, служащий в математических операциях для сохранения переноса при операциях сложения или умножения, или для указания того, что при вычитании результат отрицателен. Флаг переноса также используется в некоторых операциях сдвига.

**формат ответа**

Формат, задающий данные, требуемые в ответе при передаче данных

**Функциональный код**

2-разрядный код, служащий для ввода команд в ПК.

**цикл**

Повторяющийся цикл работы ЦПУ, включающий обработку релейно-контактной программы, обслуживание периферийного порта, обновление входов/выходов и т. Д.

**циклический регистр сдвига**

Регистр сдвига, в котором данные, выдвигаемые с одного конца, вдвигаются в регистр с другого конца.

**циклическое прерывание**

См. Прерывание по расписанию

**цифра**

Единица представления чисел, состоящая из 4 битов.

**ЦПУ**

Центральное устройство контроллера. Устройство, способное хранить программу и данные, а также выполнять команды, содержащиеся в программе. В программируемом контроллере ЦПУ выполняет программу, обрабатывает входные/выходные сигналы, осуществляет связь с другими устройствами и т. Д.

**шина**

Линия связи, служащая для обмена данными между подключенными к ней блоками.

**шина релейно-контактной схемы**

Линия на релейно-контактной схеме, идущая сверху вниз обычно с левой, а иногда и с правой стороны релейно-контактной схемы. Исполнение команд идет производится сверху вниз вдоль шины релейно-контактной схемы, которая является исходной точкой для всех командных линий.

**электрические помехи**

Случайные изменения в электрических характеристиках, таких, как напряжение, ток, что может помешать нормальной работе устройства.

**ЭСПЗУ**

Электрически стираемое ПЗУ

*Уважаемые Пользователи!*

*Данное Руководство постоянно совершенствуется.*

*В случае, если у Вас будут какие-либо замечания к данному Руководству, просим Вас сообщать о них по следующим телефонам в г. Минске:*

*017 / 229 24 22 017 / 229 28 89*

*или E-Mail: [witaly\\_z@hotmail.com](mailto:witaly_z@hotmail.com)*

**OMRON**

**OMRON EUROPE B.V.**

Представительство в СНГ

Россия, 107005 Москва

Бригадирский пер. 6

Тел. (095) 258 62 20, 258 62 21

Факс (095) 258 62 80

